Preliminary ZGRASS-32 Glossary of:
BUZZWORDS, COMMANDS, FUNCTIONS,
IDIOSYNCRASIES,
SWAP COMMANDS, SWAP FUNCTIONS,
SWITCHES, AND ESOTERICA

March 18, 1981

RESIDENT ZGRASS COMMANDS and FUNCTIONS
March 18, 1981

GRAPHICS/ ARRAYS	INPUT/ OUTPUT	MATH	PROGRAM FLOW
ARRAY	GETDISK	ARCCOS	.B
ARRAY.INT	GETTAPE	ARCSIN	.F
ARRAY.STR	INPUT	ARCTAN	GOTO
BOX	INPUT.NAME	COSINE	IF
CENTER	INPUT.STR	EXP	JUMP
CLEAR	PERSCI	INT	RETURN
CLEAR.CRT	PORT	LOG	SKIP
CLEAR.WINDOW	PRINT	LN	STOP
DISPLAY	PRINT.FORCE	POWER	TIMEOUT
ICOL	PROMPT	SINE	WAIT
LINE	PROMPT.FORCE	SQRT	
POINT	PUTDISK	TANGENT	
POINT.SNAP	PUTTAPE		
SNAP	RS232		
WINDOW	TERMINAL		
WINDOW.BOX			
WINDOW.FULL			

ERROR HANDLING	MISC.	STRING MANIPULATION	USER INFO.
LOOPMAX	COMPILE	ASCII	ADDRESS
ONERROR	CONTROL	STRING	ANYARGS
	DELETE		CORE
	EDIT		HELP
	RESTART		USEMAP
			VERSION

ZGRASS SWAP COMMANDS and SWAP FUNCTIONS
March 18, 1981

GRAPHICS/ ARRAYS	INPUT/ OUTPUT	MATH	PROGRAM FLOW
CMPARA	MUSIC*	BITS*	
COLORS	TABLET	ZAP1	
DRAG*		ZAP2	
ELLIPSE*			
FILL			
FONT			
PATTERN			
SCALE			
SCROLL			
SHRINK			
TEXT			
TXT			
WRAP*			

ERROR HANDLING	MISC.	STRING MANIPULATION	USER INFO.
GETERROR	XR	BUMP	DEBUG
		FORMAT	STATUS*
		LEN	WHATSIS
		LPAD	
		MATCH	
		REPLACE	
		SUBSTR	

* NOT AVAILABLE YET

Preliminary ZGRASS-32 Glossary of:
BUZZWORDS, COMMANDS, FUNCTIONS
IDIOSYNCRASIES,
SWAP COMMANDS, SWAP FUNCTIONS,
SWITCHES, AND ESOTERICA

March 18, 1981

Note: BUZZWORDS are common computer terms. IDIOSYNCRASIES are concepts and features peculiar to or specially modified for ZGRASS. SWAP COMMANDS and SWAP FUNCTIONS have to be gotten from disk or tape first. SWITCHES modify commands. The ESOTERICA are the advanced features for experienced programmers.

ABBREVIATION
Idiosyncrasy
you can abbreviate COMMAND, FUNCTION, VARIABLE, and MACRO NAMEs. For example:

```
PRINT 5
```

is the same as:

```
PR 5
```

This can cause confusion if you are not careful when you abbreviate NAMEs.
Example:

```
TRY1=6
TR=2
```

will cause TRY1 to be equal to 2 because TR is a valid abreviation for TRY1.
To verify this:

```
PRINT TR,TRY1
```

ADDRESS
Esoteric Buzzword
the number which corresponds to the location of data in MEMORY.

ADDRESS(NAME)
Esoteric Function
returns an INTEGER which represents the ADDRESS of the NAME.
Example:

```
SAM=5
PR ADDRESS(SAM)
```

returns a number corresponding to SAM's address in decimal

ALGORITHM
Buzzword
is a method you use to solve a problem.

AND
Buzzword

works on BITs. It makes 1's AND'ed with 1's equal to 1; and all other combinations produce 0. AND table using 2 BITs:

AND	12 00	13 01	14 10	15 11
00	00	00	00	00
01	00	01	00	01
10	00	00	10	10
11	00	01	10	11

The AND COLOR MODES are 12-15. The AND DISPLAY MODES are 3,13,23,...,133,143.

ANYARGS()
Esoteric Function

returns 0 if no ARGUMENTs left in the ARGUMENT list passed to a MACRO and 1 if there are ARGUMENTs left in the ARGUMENT list.
Example:

```
ADDEMUP=[SUM=0
IF ANYARGS()==1,INPUT A;SUM=SUM+A;SK 0
PRINT SUM]
ADDEMUP 5,10,15,20
```

ADDEMUP will add up all the arguments passed to it, and then print the total which is 50 in this case.

ARCCOS(NUMBER)
Function

returns the inverse cosine of NUMBER.

ARCSIN(NUMBER)
Function

returns the inverse sine of NUMBER.

ARCTAN(NUMBER)
Function

returns the inverse tangent of NUMBER

ARGUMENT
Buzzword

is computer talk for the stuff between commas that you give to a COMMAND, FUNCTION, or MACRO. (Actually, the first ARGUMENT has a space or '(' to its left and the last has a NEXTLINE, ';' or ')' to its right, but there are always commas in between ARGUMENTS). ARGUMENTS must be VARIABLES, NUMBERs, or EXPRESSIONS. Generally speaking, the presence of an ARGUMENT does not mean anyone is disagreeing about anything.

Note: superfluous spaces between ARGUMENTs and at the end of the line are not allowed. CTRL+Y will place a "!" at the end of each line marking the NEXTLINE so you can tell if there is an extra space between the last ARGUMENT and the NEXTLINE.

ARGUMENT LIST
Buzzword

is the list of ARGUMENTS that you give (pass) to a COMMAND, FUNCTION, or MACRO. You assign the passed ARGUMENTS to VARIABLES in a MACRO by using the INPUT COMMAND (see INPUT).
Esoteric Note:
VARIABLES are passed by NAME. Complex EXPRESSIONS (A+6-2) are EXECUTEd when they are passed. If you want to pass a VALUE, and the value is in a single VARIABLE (not an expression), use the "?" OPERATOR.
For instance:

```
A=10
PRINT A,A=100
```

will print 100,100. Since the ARGUMENTs are scanned before they get to PRINT.

```
A=10
PRINT ?A,A=100
```

will print 10,100
It is especially important to note that if LOCAL VARIABLEs are passed by NAME (no "?"), the called MACRO will not be able to access the LOCAL VARIABLE of the calling MACRO. If you must pass by VALUE, the following is an example of how to do it:

```
FEE=[a=100
FOO ?a]

FOO=[INPUT b
PRINT b*b]
```

Using "a+0" will also force evaluation for numerical VARIABLES. For STRINGs use "?" (for example, ?ABC), or CONCATENATE a null string. (ie. ABC&[])
This problem shows up in global VARIABLES too. Compare:

```
TOM=[A=100
SAM A]

SAM=[A=10
INPUT B
PRINT B*B]
```

will print 100 whereas:

```
TOM=[A=100
SAM A+0]
```

will print 10000
If you want to force passing by VALUE, use the "?" OPERATOR. ZGRASS needs to be able to pass by NAME so the ASSIGNMENT OPERATOR can be used in EXPRESSIONS and so certain FUNCTIONS (like TABLET, for example) can return more than one VALUE.

ARRAY NAME,NUMBER
Command

creates a FLOATING POINT array with elements referenced by NAME(0), NAME(1),...,NAME(NUMBER-1). ARRAYs up to four dimensions are allowed. (See definition of INDEX.)
Examples:

```
ARRAY ROOTS,10
```

will create a 10 element array referenced by ROOTS(0),....,ROOTS(9).

```
CARS=[ARRAY BUICK,100
A=0
BUICK(A)=1%320
A=A+1
IF A<100,SK -2
A=0
BOX 0,0,BUICK(A),BUICK(A+1),7
A=A+2
IF A<100,SK -2]
```

will fill an array, BUICK, with 100 RANDOM VALUES and use them to draw 50 BOXes.

```
ARRAY CHECKER,10,10
```

will create a 100 element array referenced by CHECKER(0,0),CHECKER(0,1),...,CHECKER(9,9).
See INDIRECTION for another example using ARRAY.

ARRAY.INT NAME,NUMBER
Command

creates an INTEGER array with elements referenced by NAME(0), NAME(1),...,NAME(NUMBER-1). ARRAYs up to four dimensions are allowed.
Example:

```
SHOW=[ARRAY JANE,200
A=0
JANE(A)=1%100
A=A+1
IF A<10,SK -2
CLEAR.C
USEMAP
A=0
PRINT "JANE("&A&")="&JANE(A)
A=A+1
IF A<10,SKIP -2]
SHOW
```

When you run SHOW it will first create the ARRAY JANE, then assign a RANDOM number to each element in JANE, then generate a USEMAP listing so you can see the size of JANE, and finally print out the first ten elements. If you change ARRAY JANE to ARRAY.INT JANE you will notice USEMAP lists JANE as about half as big and since JANE now holds only INTEGERs, only INTEGER VALUES are printed.
See INDIRECTION for another example using ARRAY.

ARRAY.STR NAME,NUMBER
Esoteric Command

creates a STRING array with string elements referenced by NAME(0), NAME(1),...,NAME(NUMBER-1). ARRAYs up to four dimensions are allowed. To store STRING ARRAYs on tape or disk you need to use GTSTRING/PTSTRING or GDSTRING/PDSTRING, SWAP MODULES which are not yet available.

Example:

```
ARRAY.STR ATHRUZ,26
ALPH=[I=0
ATHRUZ(I)=ASCII(I+65)
PRINT "ATHRUZ("&I&")="&ATHRUZ(I)
IF (I=I+1)<26,SK -2]
```

This MACRO will fill the STRING ARRAY ATHRUZ with the letters A-Z and print them out.

ASCII (NUMBER)
ESOTERIC Function

returns a one character STRING corresponding to NUMBER, an ASCII value. ASCII is the coding system for characters, numbers and punctuation. Refer to a standard ASCII table for specific values. The STRING COMMAND takes characters and returns their ASCII values.

Example:

```
NUMS=[K=48
ZEROTONINE=ZEROTONINE&ASCII(K)
IF (K=K+1)<58,SK -1
PRINT ZEROTONINE]
```

The ASCII values for the characters 0-9 are 49-58. This MACRO CONCATENATES the characters 0-9 and them prints them out as "0123456789".

ASSIGNMENT
Buzzword

Examples:

```
A=100
```

This assigns the VALUE 100 to the VARIABLE A.

```
LETTERS="ABCDEFG"
```

assigns the STRING "ABCDEFG" to the VARIABLE LETTERS.

```
LONGSTRING="THIS IS A VERY VERY LONG STRING
WITH NEXTLINES AT THE END OF EVERY LINE.
NOTICE YOU CAN HAVE NEXTLINES, COMMAS,
PERIODS, AND ANY OTHER PUNCTUATION EXCEPT A
DOUBLE QUOTE IN THIS CASE."
```

Note that you can assign very long STRINGs to VARIABLES.

```
NULLSTRING=""
```

A VARIABLE can have a NULL STRING as its VALUE.

```
ROOT=(-B+SQRT(B*B*A*C))/2
```

EXPRESSIONS can be assigned to a VARIABLE.
You can put ASSIGNMENTs in EXPRESSIONS:

```
TOM=[IF A<160,BOX 0,0,A=A+10,A,3;SKIP 0]
```

ASSIGNMENT OPERATOR
Buzzword

is the '=' sign.

.B
Switch

NAME.B means run NAME in the background over and over again interleaved with other .B MACROs, if any, until CONTROL+C or STOP NAME is seen. You will notice that when you .B a MACRO the ">" cursor is still there which means you can issue COMMANDs from the keyboard, EXECUTE other MACROs, .F MACROs all of which take precedence.
Example:

```
BOX 0,0,18,18,1
BOX 0,9,2,4,3
SNAP APPLE,0,4,24,24
CLEAR
ANIMATE=[DISPLAY APPLE,X=X+$X1,Y=Y+$Y1,0]
ANIMATE.B
```

will move the APPLE (a SNAPed picture element) under the control of the first JOYSTICK until further notice. Try COMPILing ANIMATE to see the APPLE move faster. Then try typing in other COMMANDs and see the .B MACRO stop while the COMMAND is EXECUTEd.

```
COMPILE ANIMATE,FASTER
FASTER.B
```

Any MACRO called by a .B MACRO will be executed as if it were a single line, that is, without interleaving with other .B MACROs.

BIT
Buzzword

is a single binary value, either 0 or 1. There are two BITs for each PIXEL on the screen. Since one BIT can specify one of two NUMBERS, two BITs can specify four NUMBERS, which is why four COLORs can be displayed on the screen at any one point. There are eight BITs in a BYTE, and sixteen BITs in a INTEGER.

BOX XCENTER,YCENTER,XSIZE,YSIZE,COLORMODE
Command

draws a filled in rectangle of the dimensions XSIZE by YSIZE centered at XCENTER,YCENTER with drawing mode specified by COLORMODE (see COLOR MODES for the 21 options).
Example:

```
BOX 0,0,40,30,1
```

draws a rectangle centered at 0,0 which is 40 PIXELS wide, 30 PIXELs high, and is drawn in COLORMODE 1. If you draw a BOX which as a whole can't fit on the screen, it will be CLIPPED to the edges of the screen. For example:

```
BOX 75,45,50,50,1
```

will put a 30X30 BOX in the upper right corner.

BUMP STRING,NUMBER
Esoteric SWAP Command

increments the ASCII code of the last non-null character in a string by a specified numeric value.

Example:

```
TEST="ABCDE"
BUMP(TEST,2)
PRINT TEST
```

prints out the string "ABCDG"

Note: BUMP does not cause the re-assignment of the STRING so:

```
TEST="ABCDE"
BARB=TEST
BUMP(TEST,2)
PRINT TEST,BARB
```

will print "ABCDG" twice. CONCATENATE BARB with a STRING to avoid this if necessary. (BARB=BARB&[])

BYTE
Buzzword

a BYTE is the amount of MEMORY needed to hold a single character. Computers generally store one BYTE at each MEMORY location. ZGRASS lists the amount of MEMORY a NAMEd thing takes up in BYTEs when you use the USEMAP command.

CALL
Buzzword

is what you do to cause the execution of a MACRO, COMMAND, or FUNCTION, that is, specifying its NAME and ARGUMENTs. ZGRASS has no CALL COMMAND since specifying a NAME plus ARGUMENTS is enough to call the MACRO, FUNCTION or COMMAND.

CENTER XCOOR,YCOOR
Command

changes the center of the screen, default of which is 0,0. See STATUS.

Example:

```
BOX 5,5,10,10,1
CENTER 80,50
BOX 5,5,10,10,1
```

CENTER will change the center of the screen to the lower left corner to allow all the X&Y COORDINATES to be positive. Notice the different positions of these BOXes.

```
CENTER 20000,20000
```

will allow displays with COORDINATES in the range:

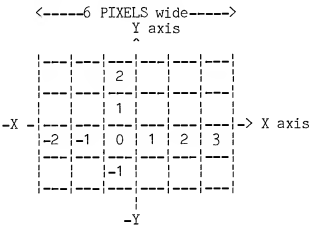
X axis 19841-20160
Y axis 19900-20101

This could be useful for roaming around large databases like the map of a city.

Use the Swap Command STATUS to find the current screen center.

CENTERING (of Graphics Primitives)
Idiosyncrasy

The centering of even numbered dimensions is biased to the lower left. The upper right hand corner of the lower left quadrant is the center pixel. For example given a BOX centered at 0,0 which is 6 PIXELs wide on the X-axis, and 4 PIXELs high on the Y-axis, the left X would be -2, bottom Y -1, right X 3, top Y 2.



You can see that the center PIXEL in this 6X4 box is located in the the upper right hand corner of the lower left hand quadrant.

CLEAR
Command

clears the TV screen (not the computer's memory). See FRAME BUFFER. RESTART clears the computer's memory.

CLEAR.CRT
Command

clears the CRT screen.

CLEAR.WIND
Command

clears the graphics WINDOW.

CLIPPING
Buzzword

refers to the action of displaying only a portion of a LINE, SNAP, or BOX if part of it exceeds the screen or window boundaries. Example:

```
BOX 60,40,50,50,3
```

will put a BOX in the upper right corner and throw away parts exceeding 80 in the X direction and 50 in the Y.

CMPARA(A1,B1)
Esoteric Swap Function

returns values depending on the comparison of two ARRAYs (usually used to compare SNAPs). The values returned are:

0 if all the BITs of A1<=A2
1 if all the BITs of A1==A2
-1 if all the BITs of A1>=A2
-2 otherwise

Example:

```
BOX 0,0,20,20,1
SNAP FIRST,0,0,20,20
BOX 0,0,20,20,3
SNAP SECOND,0,0,20,20
PRINT CMPARA(FIRST,SECOND)
```

prints 0 because all of FIRST is 01 PIXELs which are all less than or equal to all of SECOND's 11 PIXELs. If the second box were drawn in COLOR MODE 2, the result would be -2.

COLOR
Idiosyncrasy

The 256 COLORs available in ZGRASS form an abreviated spectrum. You can get four COLORs on the screen at any one point. The default COLOR VALUES are white (7), red (91), green (165), and blue (8). By using the DEVICE VARIABLES $L0 through $L3 you can change the currently available palette of 4 COLORs. The VALUE of $L0 is 7 (white). The VALUE of $L1 is 90 (red), etc. See COLOR MAP for how ZGRASS keeps track of these four COLORs.

COLOR MAP
Idiosyncrasy

The COLOR MAP is the way ZGRASS translates COLORs 0-3 into the 256 available COLOR VALUES. The hardware looks at the values of $L0-$L3 before it writes a PIXEL to the screen. If it is writing a 0 it uses the COLOR VALUE (0-255) stored in $L0. If it is writing a 1, it uses the COLOR VALUE stored in $L1, and so on. To change the COLOR MAP so 1 refers to yellow instead of red:

```
$L1=127
```

There are actually two COLOR MAPs, the $L's and the $R's. You get to the $R's by setting $HB. (see DEVICE VARIABLES)

Example:

```
CBARS=[CLEAR;A=-70;C=0;$HB=20
$R0=0;$R1=82;$R2=43;$R3=249
$L0=7;$L1=213;$L2=126;$L3=164
C=(C+1)\3+1
IF A<50,BOX A=A+20,0,20,100,C;SKIP -1]
```

This will make a set of colorbars for tuning your TV.

COLOR MODES
Idiosyncrasy

The possible values for COLOR MODES are 0-21. 0-15 are resident in ZGRASS and 16-21 are in the Swap Command COLORS. You may need to study your truth tables for PLOP, XOR, OR, AND, PRIORITY, and REVERSE-PRIORITY logical operations to really understand what's going on. Look under PLOP, XOR, etc. for their respective truth table.

COLOR MODE	Meaning:
0	PLOP with COLOR 00 (white)
1	PLOP with COLOR 01 (red)
2	PLOP with COLOR 10 (green)
3	PLOP with COLOR 11 (blue)
4	XOR screen with COLOR 0 (no change)
5	XOR screen with COLOR 1
6	XOR screen with COLOR 2
7	XOR screen with COLOR 3
8	OR with 00 (no change)
9	OR with 01 (if white or red, turn red if green or blue, turn blue)
10	OR with 10 (if white or green, turn green, if red or blue, turn blue)
11	OR with 11 (turn blue)
12	AND with 00 (turn white)
13	AND with 01 (if white or green turn white, if red or blue, turn red)
14	AND with 10 (if white or red, turn white, if green or blue, turn green)
15	AND with 11 (no change)
16	PRIORITY WRITE 01 (if white or red turn red, if green stay green, if blue stay blue)
17	PRIORITY WRITE 10 (if white, red or green turn green, if blue stay blue)
18	REVERSE-PRIORITY 10 (green and blue, turn green, red stays red, and white stays white)
19	REVERSE-PRIORITY 01 (red, green, and blue turn red, and white stays white)
20	Increment COLOR (if white turn red, if red turn green, if green turn blue, if blue turn white)
21	Decrement COLOR (if white turn blue, if red turn white, if green turn red, if blue turn green)

COLORS
Swap Module

is the SWAP MODULE which contains COLOR MODES 16-21 and DISPLAY MODES 5-144.
Example:

```
GETTAPE COLORS
COLORS
```

allows you to use COLOR MODES 16-21 and DISPLAY MODES 5-147. If you do not get and EXECUTE COLORS first you will get ERROR #26 when using COLOR MODES 16-21 and DISPLAY MODES 5-147. Do not DELETE COLORS if you intend to continue to use COLOR MODES 16-21 and DISPLAY MODES 5-147.

COMMAND
Buzzword

there are three types of COMMANDs: system COMMANDs, SWAP COMMANDs, and ones you define yourself, called MACROs. System COMMANDs are built-in and are listed by the HELP COMMAND. Swap COMMANDs function like System COMMANDS except they must first be gotten from tape or disk.

COMMENT
Buzzword

it is helpful to have COMMENTs in your MACROs to tell how they work. In ZGRASS, a line which starts with a '.' is taken as a COMMENT. You can also have COMMENTS on lines where there are COMMANDs by using a ';' and then a '.'. Examples:

```
.THIS LINE IS TAKEN AS A COMMENT
LINE 6,-70,1;.THIS LINE HAS A COMMAND AS WELL!
```

COMPILE NAME,NEWNAME
Command

takes a MACRO called NAME, and creates a compiled MACRO called NEWNAME. Compiled MACROs are larger but run faster. They cannot be stored on disk or tape.
Note: several COMMANDS; EDIT, CORE, HELP and USEMAP if included in a MACRO will cause your MACRO not to be able to be COMPILEd and you will get ERROR #59.
Example:

```
TALL=[ARRAY LONGNAME,200
INDEX=0
LONGNAME(INDEX)=SQRT(INDEX)
INDEX=INDEX+1
IF INDEX<200,SKIP -2]
```

TALL will take approximately 15.5 seconds to run.

```
COMPILE TALL,FASTER
```

FASTER will take approximately 3.5 seconds to run. The compiler figures out NAME references, SKIPs, GOTOs, and figures out OPERATORS and parentheses. You will see better improvements in compiling when you have long programs with lots of arithmetic and/or long NAMES, or lots of LOCAL VARIABLES.

COMPILing BOX COMMANDS, on the other hand, gives a less dramatic speed increase because the time is spent mostly drawing to the screen, not figuring out the ARGUMENTS.

CONCATENATION
Buzzword

is joining STRINGs together with the '&' operator.
Examples:

```
PRINT "A"&"B"&"C"  prints ABC
PRINT "A"&10       prints A10
N="MOON"
S="SHINE"
PRINT N&S          prints MOONSHINE
```

CONSTANT
Buzzword

Examples:

```
PRINT 'THIS is a constant or literal STRING'
PRINT 33.75
PRINT 1.23E17
```

- Constants, unlike VARIABLEs, never change. You can have both NUMBERS and STRINGS as constants.

CONTROL CHARACTERS
Buzzword

are single character requests you type on the keyboard by holding the key marked CTRL down (as you would the shift key) and at the same time pushing any key from A to Z. See the CONTROL COMMAND for the listing of the CONTROL CHARACTERS.

CONTROL(NUMBER)
Esoteric Function

returns the current value of the CONTROL CHARACTER identified by NUMBER. For instance, to see if CTRL+Y is on:

```
PRINT CONTROL(25)
```

if CONTROL+Y is on the answer will be 1, and, if it is off, 0.

CONTROL

CHAR.	NUM.	TYPE	DESCIPTION:
A	1	S	;Editor delete line
B	2	*	;Resets COLORs to WRGB
C	3	S	;Stop currently running MACRO(s) clear CONTROL characters
D	4	T	;Single step in MACROs on/off with CTRL+X gives single step and listing and in the Editor moves lines
E	5	S	;Editor exit and update key
F	6	T	;Small/large text font ;Editor copy lines
G	7	*	;Turn off all CTRL characters (set to 0)
H	8	S	;Editor cursor right
I	9	S	;Editor cursor down
J	10	S	;Editor cursor up
K	11	S	;Editor cursor left
T	12	L	;change speed of scroll(3 speeds)
M	13	S	;Carriage return
N	14	T	;Beep on/off for CR and Disk/Tape feedback (directory)
O	15	T	;Supress/allow printing on CRT
P	16	T	;Echo CRT on printer if available
Q	17	T	;Start/Halt printing on CRT MACRO also waits
R	18	S	;Editor delete character
S	19	S	;Editor set move pointers
T	20	S	;Editor delete move pointers
U	21	*	;Line erase
V	22	T	;Small/large character window Allows accessory RS232 input in parallel with keyboard RS232 input
W	23	T	;Twenty line mode on/off waits for CR to print 20 lines
X	24	T	;List on/off as MACRO EXECUTES
Y	25	T	;A "!" is put at the end of every ;line which has a CR
Z	26	S	;Stop MACRO in progress and accept lines till CR alone typed

TYPES:

T is a toggle switch which you can turn on (1) or off (0) by keyboard action.

S can only be set (1) by keyboard action. You can set these to any number including zero with the CONTROL NUMBER1,NUMBER2 command below.

* this CONTROL CHARACTER is not accessible through the CONTROL COMMAND.

Note: the CONTROL Characters which are used by the EDITor can be reset by using the TERMINAL Command.

CONTROL NUMBER1,NUMBER2
Esoteric Command

Like CONTROL (NUMBER) but it writes NUMBER2 in the CONTROL CHARACTER indicated by NUMBER1. Use to set CONTROL CHARACTERs in a MACRO. (Setting CONTROL CHARACTERS B,G,U to 1 doesn't do anything, however.) CONTROL CHARACTERS used in EDIT (A,E,F,H,I,J,K,R,S,T) may be used by you for your own purposes outside EDIT. Characters D,N,O,P,Q,V,W,X,Y are set to one by an odd number of user CTRL key presses and cleared to zero by even presses. The rest are set by one or more user presses and cleared by system actions.

Examples:

CONTROL 3,1;.Will cause a CTRL+C to happen programatically

CONTROL 16,1;.Will cause whatever comes out on the CRT to be printed on the printer, if available.

CONTROL 15,1;.Will cause whatever you type on the computer terminal to be not printed to the CRT until CONTROL 15,0 is EXECUTEd.

CONTROL 24,1;.Will cause listing of lines as they EXECUTE until CONTROL 24,0 is EXECUTEd.

COORDINATES
Idiosyncrasy

are the values across the X (horizontal) axis and up and down the Y (vertical) axis. The COORDINATES range from -32768 to 32767. With the default WINDOW in effect the visible X-COORDINATES range from -79 to 80, and the Y-COORDINATES range from -49 to 50. See WINDOW.

CORE
Command

tells you how much memory you have in BYTES in how many fragments. The first number is the hexidecimal ADDRESS which you should ignore. A BYTE will hold one character so if you have a MACRO on tape that is 500 BYTES long (USEMAP will give its length once it's in memory), CORE has to show a fragment with a least 500 BYTES for you to GETTAPE it without getting ERROR #27 (not enough memory space).

CORE()
Function

returns the size of the largest block of MEMORY left and also prints the CORE map. (You can supress the printing with CONTROL 15,1.)

Example:

```
A=CORE()
```

will print a list of the available memory

```
PRINT A
```

will print 4064 if this is done right after RESTART.

COSINE(NUMBER)
Function
returns the cosine of NUMBER.

CURSOR
Idiosyncrasy
is the little box over a character in EDIT. The next thing you do in EDIT (insert, delete, etc.) will be done at the CURSOR position.

DEBUG
Esoteric Swap Command
Refer to the Swap Module creation documentation, a separate package.

DELETE NAME
Command
deletes the NAME (VARIABLE, ARRAY, STRING) from memory and reclaims the memory for further use. Certain things cannot be deleted (DEVICE VARIABLES, the VARIABLES A-Z, system COMMANDS, and FUNCTIONS) so an appropriate ERROR message accompanies illegal deletion requests. Never DELETE anything that is referenced in a COMPILEd MACRO unless you have already DELETEd that COMPILEd MACRO or intend not to use it again.
Example:

```
GONE="WITH THE WIND"
```

USEMAP will tell you that there is a STRING called GONE in MEMORY.

```
DELETE GONE
```

USEMAP will now tell you that GONE is no longer in MEMORY.

DEVICE VARIABLES
Idiosyncrasy
are special VARIABLES starting with a '$' that access system features. You use them just like other VARIABLES.

VARIABLE:	Description:	Range:
Screen COLOR VARIABLES:		
$L0	COLOR 0 left	0-255
$L1	COLOR 1 left	0-255
$L2	COLOR 2 left	0-255
$L3	COLOR 3 left	0-255
(left means left half of screen set by $HB)		
$R0	COLOR 0 right	0-255
$R1	COLOR 1 right	0-255
$R2	COLOR 2 right	0-255
$R3	COLOR 3 right	0-255

('right' means right half of screen, set by $HB)

$HB	Horizontal Color Boundary	0-44
$BC	Border Color	0-3
	0 set Border to $L0	
	1 set Border to $L1	
	2 set Border to $L2	
	3 set Border to $L3	

JOYSTICK control VARIABLES:

$X1-$X4	X of JOYSTICKs 1-4	-1,0,1
$Y1-$Y4	Y of JOYSTICKs 1-4	-1,0,1
$K1-$K4	knob value of JOYSTICKs 1-4	-128 to 127
$T1-$T4	trigger value of JOYSTICKs 1-4	0 or 1

System Timers:

$Z0-$Z9 decremented by 1 every 1/60 second until 0

$TK system time in 1/60's seconds up to 60

$SC in seconds up to 60

$MN in minutes up to 60

$HR in hours up to 24

$DA in days up to 32767

$ST in seconds up to 32767

Example:

```
CLOCK=[PR $HR,':',$MN,':',$SC,':',$TK;SK 0]
CLOCK.B
```

DISPLAY NAME,XCENTER,YCENTER,DISPLAYMODE

Command

takes a SNAPped NAME and writes it at the center indicated using DISPLAYMODE. Refer to DISPLAY MODES for the details on the 74 different writing modes. (A SNAPped NAME is actually an ARRAY specially created by the SNAP COMMAND and is essentially an exact copy of an area of screen memory.) You can use DISPLAY for animation. Say there is an hat drawn at the center of the screen which fits inside a rectangle of 24X24 PIXELs. The following code will draw it, SNAP it and allow you to move it on a JOYSTICK.

```
CLEAR
BOX 0,2,14,16,3
BOX 0,-8,22,4,3
SNAP HAT,0,0,24,24
.LEAVE EXTRA WHITE AROUND FOR ERASING
MOVE=[DISPLAY HAT,X=X+$X1,Y=Y+$Y1,0;SKIP 0]
MOVE
```

Note: The largest area you can SNAP is the entire screen. (160X100)

DISPLAY MODES
Idiosyncrasy

the possible values for DISPLAY MODE are between 0 and 147. You may need to study your truth tables for PLOP, XOR, OR, AND, and PRIORITY logical operations to really understand what's going on. There are 8 logic modes, mentioned above which we combine with 15 filters (0,10,20...140) to come up with 120 DISPLAY MODES:
0,1,2,3,4,5,6,7,11,12,13,14,15,16,17,...,
140,141,142,143,144,145,146,147

Logic MODES Meaning:

0	PLOP the SNAPped NAME on the screen
1	XOR the SNAPped NAME with the screen
2	OR the SNAPed NAME with the screen
3	AND the SNAPed NAME with the screen

To use Logic Modes 4-7 get the Swap COLORS.

4	PRIORITY WRITE red(01) covers white(00) green(10) covers white and red blue(11)covers white, red and green
5	PLOP only with the colors mentioned in the filter
6	XOR only with the colors mentioned in the filter
7	OR only with the colors mentioned in the filter

FILTERS: DISPLAY only this COLOR in SNAPed NAME:

0	everything (white,red,green,blue)
10	white (00)
20	red (01)
30	green (10)
40	blue (11)
50	red and blue
60	green and blue
70	red and green
80	white and blue
90	white and red
100	white and green
110	white, red and green
120	white, red and blue
130	white, green and blue
140	red, green and blue

The equation for figuring out a specific DISPLAY MODE is:
DISPLAYMODE=LOGICMODE+FILTER
Note: Modes 8-9, 18-19, 28-29, etc do not exist. The DISPLAY MODES 4-147 are available once the SWAP Module COLORS has been brought in from tape or disk and EXECUTEd.

DRAG XCEN,YCEN,XSIZE,YSIZE,XMOV,YMOV
Swap Command

moves an area of the screen centered at XCEN,YCEN of XSIZE,YSIZE dimensions in the direction defined by XMOV,YMOV using the trailing edge of the area moved to fill in the old area.

EDIT NAME
Command

edits the MACRO specified. When you get into the editor the background will turn black, and the characters will be white. The cursor, the little blue box, marks your current position. The following is a list of controls for use in the editor.

KEY LABELED:

```
<-              ;Move cursor left
->              ;Move cursor right
^               ;Move CURSOR up
\/              ;Move CURSOR down
CTRL+DEL LINE ;Delete line
CTRL+INS LINE  ;insert a line
CTRL+DEL CHAR ;Delete a character
CTRL+INS CHAR ;insert a character
CTRL+S          ;set copy/move pointers
CTRL+T          ;clear copy/move pointers
CTRL+D          ;move
CTRL+F          ;copy
CTRL+E          ;Update and exit from
                editor
RESET           ;Exit editor without
                updating
```

CTRL+C followed by another character works just like RESET. The EDITor control keys may be changed with the TERMINAL COMMAND. The CURSOR can be moved by using JOYSTICK Knob #1 ($X1 & $Y1), The trigger will do the same as CTRL+INS CHAR. So you may want to mount your first JOYSTICK on a stand next to your terminal and use it to position the CURSOR.

ELLIPSE 1,ANGLE,XCEN,YCEN,XSIZE,YSIZE,COLORMODE
ELLIPSE 0,ANGLE,XCEN,YCEN,XSIZE,YSIZE,COLORMODE
Swap Command

draws an ellipse centered at XCEN,YCEN with XSIZE as the width, and YSIZE as the height in the specified COLOR MODE. Make the first ARGUMENT 1 to get a solid ellipse, and 0 to get just the outline. ANGLE is the tilt off the X-axis in RADIANS.

Examples:

```
ELLIPSE 1,-25,25,0,40,20,1
ELLIPSE 0,25,25,0,40,20,1
```

The first draws a solid ellipse, the second just its outline.

```
ELLIPSE 1,25,-25,.7853,40,20,1
```

draws a solid ellipse tilted off the X-axis .7853 RADIANS (45 degrees).

ERROR NUMBER
Idiosyncrasy

an ERROR NUMBER is printed on the CRT if something has gone wrong in your MACRO, or if you try to do something like dividing by zero which is not allowed. Refer to the following list for a clue to what went wrong:

ERROR #: Explanation:

```
2  ;System error - RESTART system
3  ;System error - RESTART system
4  ;System error - RESTART system
20 ;Operand (VARIABLE, Number, etc.)
   expected but not seen
21 ;Something other than a legal NAME
   on the left side of an ASSIGNMENT
22 ;Can't do this conversion, only strings
   and numbers may be converted to each
   other
23 ;Arithmetic overflow (number too big to
   convert to INTEGER or exceeds FLOATING
   POINT range)
24 ;You tried to divide by zero
26 ;COLORS SWAP MODULE needed for this
   COLOR/DISPLAY MODE
27 ;Out of memory space, DELETE something
28 ;More than 128 characters typed before
   a NEXTLINE
30 ;Too many ARGUMENTs for this COMMAND
31 ;Funny SYNTAX
32 ;Extra stuff on line
33 ;Illegal character after COMMAND name
34 ;This NAME should be a MACRO but it isn't
35 ;Can't find this NAME
36 ;More RETURNs than MACRO calls
37 ;Can't find this LABEL
38 ;This NAME can't be DELETEd for system
   integrity reasons
39 ;Not enough ARGUMENTs for this COMMAND
41 ;Illegal character in NAME (must be a
   followed by letters or digits)
42 ;Unbalanced parentheses
43 ;Number expected but you forgot it!
45 ;This NAME already exists
46 ;Illegal special VARIABLE NAME
47 ;ARRAY reference out of bounds
48 ;More than 4 dimensions specified in
   ARRAY COMMAND
50 ;No such SWITCH with this COMMAND
51 ;Fraction too small (arithmetic
   underflow)
52 ;Invalid ARGUMENT value (example:
   SQRT(-1)
53 ;EDIT only works on MACROs (STRINGS)
54 ;Only A-Z allowed in CONTROL COMMAND
55 ;Too many digits after decimal point
   (6 maximum)
```

```
56 ;Negative value not allowed here
57 ;Null STRING not allowed here
58 ;Negative ARGUMENT not allowed here
59 ;Can't COMPILE this COMMAND
60 ;Duplicate LABEL
61 ;INTEGERS only for COMPILEd SKIPs
62 ;Too many lines for COMPILEr
63 ;Illegal LABEL SYNTAX
64 ;ONERROR in LOOP
65 ;LOOPMAX exceeded
66 ;System STRING error
67 ;Too many ARGUMENTS
69 ;Must be in MACRO for this COMMAND
70 ;Can't CMPARA ARRAYs of different sizes
71 ;Transmit error over auxillary RS232 Port
73 ;No such file
74 ;Feature not implemented
75 ;Disk error
76 ;Too many SKIPs, GOTOs, IFs to
   COMPILE (max is 99)
```

EXCLUSIVE OR
see XOR

EXECUTE
Buzzword

is computer talk for doing a COMMAND, MACRO, or ASSIGNMENT. It has nothing to do with killing anything. (See CALL)

EXP(N)
Function

returns the value of e (2.71828) raised to the power N.
Examples:

```
PR EXP(2)
```

prints 7.38905

```
PR EXP(1)*EXP(1)
```

prints 7.38905

EXPRESSION
Idiosyncrasy

is:
1. a CONSTANT (12, 'foo', for example)
2. a NAME (TOM, $X1, POOHBAH, for example)
3. a combination of OPERATORs and CONSTANTs or VARIABLEs (+6, ?B, -ABC, FF+1, 'tom'&'sam', Beer*4, for example).
4. a FUNCTION or MACRO call (SIN(a)+COS(b)), MAX(k,F+E,Beer), etc).

Expressions can be simple or complex. Actually anything syntactically correct ZGRASS is an EXPRESSION. Arithmetic EXPRESSIONs result in numbers being generated and are a mix of arithmetic OPERATORs (+,-,/,\,*,?,&&,!!), parentheses, numbers, and VARIABLEs. STRING EXPRESSIONs are a mix of STRING OPERATORs

(",',[,],{,},&,@,?) and STRING VARIABLEs. FUNCTIONS which return NUMBERS or STRINGs can also be parts of EXPRESSIONs. ZGRASS attempts to convert NUMBERS to STRINGS and STRINGS to NUMBERS when it can, so a STRING like ABC='1234' can legally be used in PRINT ABC+ABC or PRINT ABC&ABC, and so on. COMMANDS are EXPRESSIONS too. Most return the value 1 but some, like ANYARGS, SINE, RETURN, can return other values as well. The basic idea is to combine small EXPRESSIONS to make larger ones. Examples:

```
A
A+1
A+B*C
(A+B)*c
SIN(ABC)+COS(ABC)
C=A+BOX(10,10,20,30,5)
etc.
```

.F

Esoteric Switch

is a way of telling a MACRO to EXECUTE every 1/60 second. Such MACROs should be short since they take precedence over regular and .B MACROs.

Example:

```
TIMESUP=[timer=timer+1
IF timer==180,PRINT '3' SECS ARE UP';timer=0]
TIMESUP.F
```

Unfortunately, unless COMPILEd, this takes about 6.2 seconds to do.

FILES

Buzzword

is what things stored on disk or tape are called. FILENAMES are the NAMEs of FILES, of course.

FILL X,Y,FILLCOLOR

Swap Command

is used to fill a bounded area with a solid color. X,Y are the COORDINATES of a point interior to the boundary. FILLCOLOR can be 0,1,2,3 referring to the four COLORs $L0-$L3 (and $R0-$R3 IF $HB is set). Refer to PATTERN for FILLing areas with a pattern.

Example:

```
BOX 0,0,40,40,1
BOX 10,10,20,20,0
BOX -10,0,18,38,0
BOX 9,-10,20,18,0
```

Creates an L shaped area using red (color 01).

```
FILL 5,-10,3
```

will fill the area bounded by the red outline with blue starting at the point 5,-10.

FLOATING POINT
Buzzword

is computer talk for numbers bigger that 32767 and smaller than -32768 (16 BIT INTEGER range). Numbers outside this range and those with decimal points must be stored and computed specially for esoteric computer reasons. The trade-off is that the range of the numbers available for floating point calculation becomes enormous but the accuracy starts to slip after a while. Fractions are always converted to FLOATING POINT. The name, by the way, comes from the decimal point floating around according to the POWER of ten the number has to be raised to be able to print it out to six digits of accuracy. It is also called 'scientific' notation, and, if you are not a scientist or engineer, you will probably not need to worry about it. You can convert to whole numbers with the INT FUNCTION.

FONT STRING,ARRAYNAME,SNAPNAME,YOFFSET,LEFTX,RIGHTX
Esoteric Swap Command

is used to create and maintain ARRAYS of characters or symbols to be used with the TEXT COMMAND. Each time it's used, the FONT COMMAND adds one character or symbol into a FONT ARRAY if it has not been previously defined in that ARRAY or replaces it if it has.
The ARGUMENTs are:

STRING is a single character. This character is used to identify this entry in the ARRAY. When this character is used in a STRING in the TEXT COMMAND, the corresponding character or symbol in the 'SNAPNAME' is displayed on the screen.

ARRAYNAME is the NAME of the FONTARRAY. If this NAME already exists, the character and SNAP are added to it, replacing a previous entry having the same identifying character, if necessary. If the NAME doesn't exist, it's created.

SNAPNAME is the NAME of the SNAP to be copied into the FONTARRAY. This can be a SNAP of any character or symbol, of any size or COLOR. If it is really large, you can't have many in the FONTARRAY before you run out of space.

YOFFSET is a positive or negative INTEGER or zero. This number along with the Y-COORDINATE in the TEXT COMMAND, is used to determine the Y-COORDINATE used in displaying this character. A negative number drops the character below the line of text, a positive number raises it. This option is used for characters such a lower case g or p, which should drop below the line of text or subscripts which should go up some.

LEFTX
RIGHTX are numbers from 0 to 4. They identify the type of the left or right edge of a character. The type of the right edge of one character, and the left edge of the next, are used in the TEXT COMMAND to look up a horizontal spacing value in a two-dimensional ARRAY.
For example:

RIGHT \ LEFT	0	o	/	\	1
0	*	*	*	*	*
o	1	2	3	4	5
/	2	3	4	5	6
\	3	4	5	6	7
1	4	5	6	7	8

The value found represents a number of pixels. This value, along with the horizontal spacing constant given in the TEXT COMMAND, are used to determine the horizontal spacing between characters. The TEXT COMMAND has a built-in ARRAY with all entries of zero. Users may create their own ARRAYS and use them in the TEXT COMMAND to override the built-in ARRAY. A zero in the row column means use the default spacing. Typically you would define less space between two o's than between two l's or two m's.
Example:

Char	LEFTX	RIGHTX
O	3	3
A	4	4
L	4	1
T	1	1
C	2	3
G	3	2
Y	1	1

The spacing between "L" and "O" would be 4
The spacing between "O" and "L" would be 7

FORMAT(FLOAT,NUMBER)
Esoteric Swap Function
returns a pointer to a STRING representing a FLOATING POINT value with a NUMBER (less than or equal to six) of digits after the decimal point. ERROR #55 returned if number of digits past the decimal point is greater than six or the FLOATING POINT value is greater than 10**10.
Example:

```
PR FORMAT(10/4,3)
```

prints out the STRING "2.500" See LPAD for another example.

FRAME BUFFER
Buzzword

is used to store the images on the screen. Each PIXEL on the screen is represented by 2 BITs at a location in MEMORY. Changing that MEMORY location will change a specific PIXEL on the screen. There is 4K of screen RAM which maes the FRAME BUFFER in ZGRASS have a RESOLUTION of 160 by 100 with 2 BITs per PIXEL.

FUNCTION
Buzzword

is a COMMAND or MACRO that returns a value and is used as part of an EXPRESSION. Actually all COMMANDs and MACROs return values of 1 unless something else is specifically returned. Lots of programming languages use the term FUNCTION so we use it here as a gesture towards programmer solidarity.

Examples:

```
GREED=SIN(AVARICE)
FUNNYARRAY(MAX(A,B,C)=-9999
```

MAX is taken as a user defined FUNCTION which returns the largest of three numbers. See the RETURN Command for how MAX is written.

GETDISK NAME
Command

gets the NAMEd file from disk. May be a MACRO, ARRAY or 16K screen dump (see PUTDISK and PERSCI COMMANDs)

GETERROR(NUMBER)
Esoteric Swap Function

if NUMBER==0, returns the ERROR number that last occurred. Usually used in conjunction with ONERROR to figure out programatically what ERROR condition arose. Cannot be used outside of the MACRO in which the ERROR occured.

if NUMBER==1, returns a number corresponding to the place where the ERROR occurred on the line. 1 refers to the COMMAND name, and 4 would tell you that ARGUMENT number 3 (the fourth word) was where the ERROR was found. Used with ONERROR.

if NUMBER==2, returns the COMMAND line in ERROR as a STRING. It can be used in conjunction with GETERROR(1) to pinpoint the part of the COMMAND in ERROR and point it out friendly-like to the user of your MACRO.

Example:

```
BAD=[ONERROR 1
BOX 0,0,"!",1,3
PRINT "OK"
RETURN
1 PR "ERROR #"&GETERROR(0),"IN ARGUMENT",
GETERROR(1)-1,"ON LINE:",GETERROR(2)
RETURN]
```

will catch the ERROR ("!" is an invalid INTEGER) and print out:

```
ERROR #22 IN ARGUMENT 3 ON LINE: BOX 0,0,"!",1,3
```

GETTAPE FILENAME
Command

gets the FILENAME from tape. May be a MACRO, ARRAY, or a 4K screen dump (see PUTTAPE). If you press CTRL "N" before you

```
GETTAPE FILENAME
```

you will get a complete directory listing of everything else which is on the tape. If a read error occurs, the next copy will be read. (see PUTTAPE). Use BREAK to prematurely stop GETTAPE.
Example:

```
GETTAPE FOOD
```

will search through the tape until it finds FOOD, then print out:

```
STRING NAME: FOOD
LENGTH: NUMBER (IN BYTES)
A DESCRIPTIVE MESSAGE ABOUT THE FOOD
```

If it reads a copy of the file which has errors it will print out '***BAD READ***', and look for another copy. Switches:

.ERR accept the file even if an error is read
.ANY get the next file whatever its NAME is on tape and read it in with the NAME you specify.

GOTO LABEL
Command

causes the line which begins with LABEL to be EXECUTEd next. LABELs begin with numbers.
Examples:
These are valid LABELs:

```
10
1NOW
2small
30000
```

Example:

```
SQUARES=[A=80
1AGAIN BOX 0,0,A,A,A/10
IF (A=A-10)>0,GOTO 1AGAIN
PRINT "IS THIS ART?"]
```

HELP

gives a list of the resident COMMANDS and FUNCTIONS available along with their ARGUMENTs. Use CTRL+W to get 20-line mode and then press RETURN to see the first 20 lines. Press RETURN again for more.

ICOL ARRAYNAME
Esoteric Function

uses the values in ARRAYNAME, an INTEGER ARRAY, to cause an interrupt on one of the vertical lines of the screen. ICOL uses 10 elements in the ARRAY at a time. The last element of the ARRAY must be -50. ICOL can be cancelled by hitting CTRL+C. The following is a list of what each of the elements in each 10 element section needs:

LOCATION: VALUE:

0	Next line to interrupt at (-49 to 50)
1	Color to correspond to 00 Right
2	" " 01 Right
3	" " 10 Right
4	" " 11 Right
5	" " 00 Left
6	" " 01 Left
7	" " 10 Left
8	" " 11 Left
9	Place for Left/Right split
10	next line to interrupt at or -50 if end of the ARRAY

Line 50 corresponds to the top of the screen and line -49 to the bottom. 00 left to 11 Right, and the right/left split are defined for the area above the interrupted line (or the 0,10th, 20th etc element in the ARRAY. Failure to use an INTEGER ARRAY wipes out the system. Be careful with this COMMAND.

Example:

```
THREE=[BOX 0,0,130,95,1
BOX 0,0,100,80,2
BOX 0,0,60,65,3
$HB=22
ARRAY.INT SKY,21
SKY(0)=25
SKY(1)=10;SKY(2)=20;SKY(3)=30;SKY(4)=40
SKY(5)=50;SKY(6)=60;SKY(7)=70;SKY(8)=80
SKY(9)=17
SKY(10)=-25
SKY(11)=110;SKY(12)=120;SKY(13)=130;SKY(14)=140
SKY(15)=150;SKY(16)=160;SKY(17)=170;SKY(18)=180
SKY(19)=24
SKY(20)=-50]
```

THREE makes a graphic using all four COLORs, then sets $HB to the screen center so both COLOR MAPs can be seen, and finally loads the ARRAY SKY with VALUEs to be used with the ICOL Command.

```
ICOL SKY
```

divides the screen into 3 sections by causing INTERRUPTS at 25 and -30 (which correspond to values on the Y axis), and right/left divisions at 17 and 24 (which correspond to values on the X axis). The top section has a range of pink and blue colors, the middle green and brown, and the bottom has the default COLORs.

IF CONDITIONAL, COMMAND
Command

if the CONDITIONAL is satisfied the COMMAND following is EXECUTEd. Otherwise, control is skipped to the next line. A CONDITIONAL is a EXPRESSION which evaluates to 0 (false) or 1 (true). Expressions using RELATIONAL OPERATORS evaluate to true or false, and the rest of the line (including ';'s) is EXECUTEd if the condition is true. Anything that evaluates to 0 or 1 can be used as part of an IF statement.
For example:

```
IF A==10,PRINT A;.will print the value of A
if it is equal to 10

FIXUP=[PR "I'M YOURS"]
IF 1,FIXUP;.this will always happen

IF FLAG,B=C+D;.this will happen if FLAG==1

IF SIN(BRADIANS)*1.25<=.7,DRAW
```

The last example shows that complex EXPRESSIONS are allowed in an IF statment. Note: that "equals" as a RELATIONAL OPERATOR is "==" and a single "=" is the ASSIGNMENT OPERATOR even in IF statements. For example:

```
IF A=B,FOO
```

EXECUTES FOO as long as B#0.

INDEX
Buzzword

is the NUMBER indicating which ARRAY element is being picked. The INDEX in ABC(4) is 4. ARRAYs can have multiple indices if they are multidimensional, for example CHECKERBOARD(8,8), which has indices (0,0),(0,1),...,(7,7) allowing 64 elements.

INDIRECTION
Buzzword

allows one NAME to hold another NAME as a STRING to be used as a reference. '@' is the indirection OPERATOR. Examples:

```
TOM=12
SAM="TOM"
PRINT @SAM
```

this prints 12

```
MKARRAY=[PR "ARRAY NAME PLEASE"
INPUT.STR ANAME
PR "HOW MAY ARRAY ELEMENTS?"
INPUT n
ARRAY @ANAME,n
PRINT ANAME,"HAS ELEMENTS 0 TO",n-1
TMP=@ANAME
i=0
PROMPT ANAME&"("&i&")=?"
INPUT q
TMP(i)=q
IF(i=i+1)<n,SK -3
PRINT "ARRAY",ANAME,"HAS THE VALUES:"
i=0
PRINT ANAME&"("&i&)="&TMP(i)
IF (i=i+1)<n,SK -1]
```

When you execute MKARRAY, first it makes an ARRAY with ANAME of size n, then the user inputs values for each ARRAY element, and finally the contents of the ARRAY is printed out.
The detail to notice is:

```
TMP=@ANAME
```

This is a shortcut for dealing with ARRAY elements in a general program, so that each element can be accessed by TMP(0),TMP(1),...,TMP(n-1). We could skip the assignment of @ANAME to TMP and instead build a string:

```
@(ANAME&"("&1&")")
```

which is the same as

```
TMP(1)
```

Unfortunately the building of strings through CONCATENATION is very time consuming.

INFINITE LOOP
Buzzword

is a LOOP which has no intention of ever stopping. Such a LOOP is in error if you want the MACRO it's in to stop or are using it as a FUNCTION which is supposed to return a value. It can be useful, though, as a MACRO run under .B or .F mode or something you want to get out of by using CTRL+C. The LOOPMAX COMMAND can be used to catch infinite loops.

INPUT NAME1,NAME2,...,NAMEN
Command

gets the VALUE from the user or the ARGUMENT list passed to the MACRO and stores the VALUE as a number in NAME.
Examples:

```
ABS=[INPUT a
IF a<0,RETURN -a
RETURN a]
PRINT ABS(-10)
```

prints out 10

```
PRINT ABS(10)
```

prints out 10

```
ASK=[PROMPT "WHAT'S YOUR AGE?"
```

```
    INPUT AGE
    PRINT "YOU ARE",AGE*12,"MONTHS OLD AT LEAST"]
```

if EXECUTEd by typing:

```
    ASK 33
```

the PROMPT is suppressed.
if EXECUTEd by typing:

```
    ASK
```

the PROMPT is printed, and you have to supply the ARGUMENT by typing it in.
Note: if you are passing a VARIABLE (as opposed to the numbers being passed above), make an EXPRESSION of it by adding 0 or using the "?" OPERATOR so its VALUE is passed rather than its NAME. This is particularly important when passing LOCAL VARIABLEs and applies to ARRAY references as well.

INPUT.NAME NAME
Command

gets a STRING of characters from the user or the ARGUMENT list passed to the MACRO and checks it for valid SYNTAX, and then puts it into NAME as a STRING.
Example:

```
    WHO=[PROMPT "TYPE YOUR FIRST NAME:"
    INPUT.NAME NAME1,NAME2,...,NAMEN
    PRINT NAME,"IS A FUNNY NAME!"]
```

Note: Do not use INPUT.NAME to pass VARIABLEs to called MACROS if it is the value of the VARIABLE you want to pass. Use INPUT.STRING to pass a STRING in a VARIABLE to a called MACRO.

INPUT.STRING NAME1,NAME2,...,NAMEN
Command

gets a STRING of characters and then puts it into NAME. This option is good for reading an entire line from the terminal, including commas. It must also be used to pass a STRING with commas or spaces as an ARGUMENT, in which case it should be enclosed in quotes or other STRING delimiters.
Examples:

```
    MAILINGLIST=[PROMPT "TYPE IN A NAME, ADDRESS,
    AND PHONE # FOLLOWED BY A BLANK LINE"
    CR={
    }
    PROMPT "MORE:"
    INPUT.STR INFO
    IF INFO#{},LIST=LIST&INFO&CR;SK -2]
```

Note: when passing LOCAL STRING VARIABLEs to MACROS, make EXPRESSIONS out of them by CONCATENATing them with a null string or by using the "?" OPERATOR in front of the NAME so that the VALUE of the STRING is passed rather than the NAME of the STRING.

INT(NUMBER)
Function

FUNCTION which returns the INTEGER part of a FLOATING POINT number. INT(5%8) will give 5, 6, or 7 without the fractional part, for example.

INTEGER
Buzzword

An integer in ZGRASS is a number between 32767 and -32768. It is very easy for the computer to store and deal with numbers in this range so they are used often. Fractions and decimal points are not allowed in INTEGER arithmetic.

INTERRUPT
Esoteric Buzzword

The ZGRASS System is programmed to EXECUTE a chunk of special code every 1/60 of a second. The ICOL COMMAND allows this frequency to be changed somewhat.

ITERATION
Buzzword

is the process of solving things by doing LOOPS. Typically, in computing, ITERATION means doing things incrementally. For instance, a computer would probably walk over to the wall by accurately measuring the distance between it and the wall, computing the exact number of steps needed, and then it would take a step, see if all the steps it had to take were taken yet, and take another if not. If it made a mistake, it might crash into the wall. People, of course, do things through feedback, and often you can program that way with computer systems that are significantly better connected to you than the average payroll-check stomper (like ZGRASS is of course). To draw 100 RANDOM sized BOXes on the screen you could type in 100 different BOX COMMANDs, or write a MACRO which would do it. For example:

```
SQUARES=[B=0
BOX -70%70,-40%40,1%50,1%30,1%8
IF (B=B+1)<100,SK -1]
```

JOYSTICK
Idiosyncrasy

is the gadget with the knob and the trigger that is connected to the ZGRASS machine. You can have up to four joysticks. The first one's knob is known as $K1, its X value as $X1, its Y value as $Y1, and its trigger value as $T1 (see DEVICE VARIABLES).

JUMP ADDRESS
Esoteric Command
Refer to the Swap Module creation documentation, a separate package.

LABEL
Idiosyncrasy
GOTO 1THIS causes ZGRASS to move to whatever line begins with the LABEL 1THIS. LABELs in ZGRASS start with numbers to differentiate them from NAMES which cannot start with numbers. LABELs also cannot contain punctuation. You can't have one GOTO in a MACRO go to a LABEL in another MACRO.

LEN(STRING)
Esoteric Swap Function
returns the length of a character STRING. If the ARGUMENT is a null STRING, 0 is returned.
Example:
PRINT LEN("abcdef")
prints the VALUE 6

LINE XCOORDINATE,YCOORDINATE,COLORMODE
Command
draws a line from the previous line endpoint used in the current MACRO to the endpoint specified by the XCOORDINATE and YCOORDINATE in the COLORMODE indicated. LINE X,Y,4 will move the endpoint without drawing anything and can be used to set the first endpoint if you do not want the first LINE to start at (0,0). (See COLOR MODES) Each MACRO has its own place to store the last endpoint used and it is set to zero when the MACRO is called.
Example:
LINE 50,-30,1
draws a line from 0,0 to 50,-30.
LINE -70,20,2
draws a line from 50,30 to -70,20.
LINE 40,40,4
LINE 40,-40,3
LINE -40,-40,3
LINE -40,40,3
LINE 40,40,3
draws a blue 4X40 rectangle outline.
ZIGZAG=[LINE -79%80,49%50,0%15;SK 0]
ZIGZAG will draw RANDOM lines of different COLORs all over the screen

LOCAL VARIABLE
Esoteric Idiosyncrasy
a VARIABLE which starts with a lowercase (a-z) letter. LOCAL VARIABLEs are known only to the MACRO they are in and are deleted automatically when the MACRO returns. They help save memory and are really useful in in .B, .F, and RECURSIVE MACROs.

LOG(NUMBER)
Function

returns the logarithm base 10 of the NUMBER.

LOGICAL OPERATOR
Buzzword

returns a truth value (0 or 1). ZGRASS has logical "AND" and "OR". The "AND" OPERATOR is '&&'. The logical "OR" OPERATOR is '||'. They are useful in many situations, one of which is combining conditionals in IF statements. Examples:

```
BEEPTHEJEEP=[CONTROL 14,1]
IF A==10&&B==20,BEEPTHEJEEP;.done if A is 10
and B is 20
IF A==10||B==20,BEEPTHEJEEP;.done if either
is true
```

LOOP
Buzzword

is a series of COMMANDS done over and over until a conditional in an IF statement is true, or if the MACRO is running in .B or .F mode and CTRL+C is pressed. If the condition is never true or if it is missing, the loop is called an INFINITE LOOP. LOOPs in ZGRASS are constructed with GOTOs and SKIPs or with .B and .F. You can always get out of a LOOP with CTRL+C. CTRL+Z allows you to get out of a LOOP to do something and then get back in by pressing the RETURN key. A loop is an example of ITERATION.
Examples:

```
INFINITELOOP=[PRINT A=A+1;SK 0]
```

is a loop which will not stop because it doesn't have an end condition. CTRL+C will stop it.

```
LOOPWHICHSTOPS=[A=0
PRINT A
A=A+1
IF A<10,SKIP -2]
```

LOOPWHICHSTOPS prints 0 through 9 and stops.

LOOPMAX NUMBER
Esoteric Command

allows you to catch INFINITE LOOPs by setting a maximum for the NUMBER of SKIPs and GOTOs that can occur before ERROR #65 is caused. MACROs which have LOOPMAX commands in them cannot be COMPILED.
Example:

```
TEST=[CONTROL 1,0;.SET CTRL+A TO ZERO
PRINT "HIT CTRL+A"
ONERROR 1SLOW
LOOPMAX 100
IF CONTROL(1)#1,SK 0
RETURN
1SLOW PRINT "YOU DIDN'T HIT CTRL+A FAST
ENOUGH!"]
```

LN(NUMBER)
Function
returns the natural log of NUMBER.

LPAD(STRING,CHARACTER,FIELDWIDTH)
Esoteric Swap Function
returns a pointer to the STRING padded on the left with a specified CHARACTER so that it fits within a given FIELDWIDTH.
Examples:

```
PR LPAD("ABC","*",6)
```

prints out the STRING "***ABC"

```
PR LPAD("EXAMPLE","*",5)
```

prints out the STRING "AMPLE"

```
LEFTX=[A=2
PR LPAD(A,'X',5)
A=A*10;IF A<=20000,SK -1]
```

takes each VALUE of A and pads it on the left with X's until each number is printed in a field of 8 characters. Usually used with blanks, not X's.

```
XXXX2
XXX20
XX200
X2000
20000
```

Given JOHN is an ARRAY with 20 numbers representing USA money.

```
DOLLARSANDCENTS=[TOTAL=0
A=0
TOTAL=TOTAL+JOHN(A)
PRINT LPAD(FORMAT(JOHN(A),2),' ',20)
A=A+1
IF A<20, SKIP -2
PRINT LPAD('-','-',20)
PRINT "TOTAL=",LPAD(FORMAT(TOTAL,2),' ',13)]
```

MACRO
Idiosyncrasy

is a STRING that contains legal ZGRASS COMMANDS. Most programming languages call such things 'programs' or 'subroutines'. MACROs are user-defined COMMANDS. You can pass ARGUMENTs to MACROs with the INPUT COMMAND and return values with the RETURN COMMAND. You define a MACRO just like you define a STRING (with an ASSIGNMENT to a NAME or by using EDIT).

MATCH(OTEXT,MTEXT,LOWER,UPPER)
Esoteric Swap Function

Search for the occurence of MTEXT, a STRING, within a specified range of OTEXT, another STRING. If a MATCH is found, the returned displacement value is relative to the beginning OTEXT, the first character being the 0th one. -1 is returned if a MATCH was not found within the specified limits. The search for a MATCH may proceed from either direction. If UPPER is greater than or equal to LOWER a forward search is made. If UPPER is less than LOWER a backward search is made. (That is, the characters are still matched left to right bu the pointer backs up on failure to match instead of advancing.) MTEXT does not necessarily have to contain all the characters of the desired MATCH but rather, may use the following expression symbols:

?	(wild card) MATCH any one character
*	MATCH all characters
*text	MATCH all characters preceding actual text
text*	MATCH text and all remaining characters following text
text1*text2	MATCH all characters between text1 and text2
[chars]	MATCH first occurance of any one of the characters with the '[','']'s. All the expression symbols lose their special meaning when appearing within square brackets.
[char-char]	MATCH any character within the range specified. [0-9] is the same as specifying [0123456789]. The minus sign loses its special meaning when specified as first or last character within the square brackets.

\ ignore following character special meaning

¦ anchor MATCH to beginning or end of OTEXT depending on whether the anchor symbol occurs first or last within MTEXT

Examples:

	ANSWER:
PR MATCH("VACATION","CAT",0,7)	2
PR MATCH("VACATION","CAT",0,3)	-1
PR MATCH("ABABCDAB","?A",0,10)	1
PR MATCH("ABABCDAB","?A",4,10)	5
PR MATCH("ABABCDAB","?A",10,0)	5
PR MATCH("ABABCDAB","?A",4,0)	1
PR MATCH("SIGNAL","*",0,20)	0
PR MATCH("WHAT TIME?","ME\?",0,10)	8
PR MATCH("SIGNAL","*",20,0)	0
PR MATCH("SIGNAL","*",3,20)	3
PR MATCH("THIS IS A TEST","[AEIOU]",3,7)	5
PR MATCH("THIS IS A TEST","[A-H]",15,0)	11
PR MATCH("GRAPHICS","¦",0,10)	7
PR MATCH("COMPUTER GRAPHIX","G*X",5,20)	9

MEMORY

Buzzword

is computer storage which is divided into BYTES. ZGRASS has 64K BYTES of MEMORY. 16K is ROM (Read only memory) where the resident code for ZGRASS is stored. 16K is Screen RAM (Random access memory that feeds the TV screen). 32K is RAM used to store MACROS, ARRAYS, SWAP MODULES, SNAPS, and VARIABLES. USEMAP shows usage of the 32K RAM. CORE tells you how much of the 32K RAM you have free.

MUSIC

Esoteric Swap Command

See the separate documentation on MUSIC.

NAME

Idiosyncrasy

is any set of symbols starting with a letter that has a VALUE (TOM=5, SAM="HOWDY", for example) or an ARRAY of VALUEs (ARRAY WOMEN,13, for example). A NAME must start with a letter (or '$') and has only letters and numbers (0-9) and '$'s in it. The rule is that a STRING is not a name if it starts with a number. In this case it is either a NUMBER or a LABEL (LABELs must be the first thing on a line, of course). If it starts with a letter, it is a NAME. Any kind of punctuation ends the NAME. A NAME is also an EXPRESSION, although a very simple one. NAMES joined together with numbers and other NAMEs using punctuation (+,-,/,*,(,),etc.) are EXPRESSIONS. If a NAME begins with a lowercase letter it is LOCAL and is known only to the MACRO in which it occurs.

NEXTLINE
Idosyncrasy

is the code ZGRASS uses to represent the end of a line. It is generated by the RETURN key. Sometimes it is known as the 'carriage return' or 'CR' from the old days or 'RETURN' on most keyboards (not to be confused with the RETURN COMMAND, of course). This character is at the end of every line in a MACRO except possibly the last. It is also the key which tells ZGRASS you are finished typing in the line you have been typing. If you hit CTRL+Y, and then list out a MACRO you will see a '!' marking the position of each NEXTLINE. NEXTLINE also advances the 20-line printout mode started by CTRL+W.
Note: you can not have any spaces before the NEXTLINE. CTRL+Y is good for verifying that no spaces exist between the last character on the line you've typed and the NEXTLINE.

NUMBER
Buzzword

Examples:
1778
1.5
-44.3
3.5E6 (3.5 million)
-2E-9 (-2 trillionths)

NUMERIC VARIABLE
Buzzword

is a VARIABLE which has a NUMBER as its value. USEMAP will tell you it is a NUMBERNAME.

ONERROR LABEL
Esoteric Command

sets up a transfer to LABEL when an ERROR occurs. You can turn off ONERROR by specifying no LABEL (ONERROR by itself turns the normal ERROR CODES back on). You normally put ONERROR LABEL before a statement that is likely to cause an ERROR. You can only have one ONERROR setup per MACRO at a time but you can change it in the MACRO anytime. See LOOPMAX and GETERROR for examples of ONERROR.
Note: this COMMAND precludes you from EXECUTing a MACRO NAMEd "ONE" due to the ABBREVIATION POLICY. This is a common mistake.

OPERATOR
Idiosyncrasy

is what glues NUMBERs and NAMEs into EXPRESSIONS. OPERATORs take the values they operate on and return a single value. Each OPERATOR has a precedence, that is, a pecking order for evaluation.

OPERATOR:	MEANING:	PRECEDENCE:
@	indirect	9
?	value	9
-	unary minus	8
+	unary plus	8
%	random	7
/	division	6
\	modulus	6
*	multiply	6
+	add	5
-	subtract	5
&	concatenate	5
==	equals	4
<	less than	4
>	greater than	4
<= or =<	less than or ==	4
>= or =>	gr. than or ==	4
# or <>	not equals	4
&&	logical AND	3
||	logical OR	2
=	assign	1
()	parentheses	0

OPERATORS with higher PRECEDENCE are done before ones with lower PRECEDENCE and ones with equal PRECEDENCE are done from left to right. Examples:

2+3*4 equals 14 because '*' (6) has a higher precedence than '+' (5) so the multiplication is done first.

(2+3)*4 equals 20 because '()' (0) have the lowest precedence so everything inside them is done first.

OR
Buzzword

works on BITs. It makes BITs OR'ed with 1's equal to 1, and leaves BITs OR'ed with 0's the same as they were. OR table using 2 BITS:

OR	8	9	10	11
00	00	01	10	11
01	01	01	11	11
10	10	11	10	11
11	11	11	11	11

The OR COLOR MODES are 8-11. The OR DISPLAY MODES are 2,7,12,17,22,27,...,132,137,142,147.

OVERFLOW
Buzzword

is what happens when the range of a CONSTANT, VARIABLE, or EXPRESSION is too large or too small. For instance, many DEVICE VARIABLES represent a single BYTE of information which gives a range of 0-255 or -128 to 127. Exceeding this range causes WRAP-AROUND so 256 is actually 0, 257 is 1, 258 is 2. INTEGERs overflow after 32767 and under -32768, which causes ERROR # 23.

PATTERN X,Y,XOFFSET,YOFFSET,SNAPNAME
Swap Command

like FILL but uses PIXELs out of a SNAPNAME to fill within a bounded area. X,Y indicate the starting point for the pattern fill. The area is filled with SNAPNAME as if its lower left corner were positioned at 0,0. XOFFSET, and YOFFSET are used to change this orientation. The following example illustrates the use of the pattern fill with and without offsets.

Example:

```
OPART=[CLEAR
BOX -65,0,12,12,3
BOX -65,0,9,9,2
BOX -65,0,6,6,1
BOX -65,0,3,3,0
SNAP SQR,-65,0,12,12
BOX 0,0,90,50,1
BOX 0,0,88,48,0
BOX 20,0,26,20,1
BOX 20,0,24,18,0
BOX -20,0,26,20,1
BOX -20,0,24,18,0]
OPART
PATTERN 20,0,0,0,SQR;.NO OFFSET
PATTERN -20,0,7,-4,SQR;.OFFSET ON X AND Y
PATTERN 0,0,0,0,SQR
```

PERSCI STRING
Command

is used to send commands to the diskette controller. STRING is made up of a single command letter and followed by one or more command paramenters. ZGRASS has DRIVE 0 as its default. FILENAMES below may be suffixed with /0 to indicate DRIVE 0 or /1 to indicate DRIVE 1.

Command	Syntax:	Function:
Copy	[CFILE1 FILE2]	copies files matching FILE1 to the same or different diskette, optionally renaming according to FILE2
Delete	[DFILE]	deletes all files matching FILE

Gap [G/DRIVE] Compresses allocations on the DRIVE to eliminate gaps (use 0 or 1 for DRIVE)

Kill [KLABEL 6] or [KKLABEL 6] Initializes diskette with LABEL as a title and interleaving factor 6

Mode [M/DRIVE] sets the current drive (use 0 or 1 for DRIVE)

Name [NFILE1 FILE2] renames file FILE1 to FILE2

Query [QFILE] Reports directory information files matching FILE (use [Q*] to get a complete directory listing)

The special characters "?" and "*" may be used to make a file reference match a number of different files. The "?" is used as the wild-card character which matches any character in the corresponding position in a file reference.

Example:

```
PERSCI [QWOM????.G???]
```

matches all the following file references:

```
WOMYN.GREAT
WOMEN.GLUM
WOMBAT.GET
```

The character "*" is used to denote that all character positions to the right are wild-cards unless otherwise specified. The following examples illustrate the flexibility which this facility provides:

```
MONITOR.*     all files with the name MONITOR
*.ZGR         all files with version .ZGR
C*            all files starting with C
*             all files on diskette
```

ZGRASS files may have only 8 characters in them if you want to store them on disk. Files with over 8 characters will have a "." put in them which you can ignore. For more information about the PERSCI commands, read your PERSCI 1077 Manual.

Examples:

```
PER [CTOM T1/1]
```

copies TOM from disk drive 0 to 1 renaming it T1.

```
PER [C* */1]
```

copies all the files from drive 0 to drive 1.

```
PER [G]
```

reclaims space lost when deleting files, on which ever disk is the default.

```
PER [DAB*]
```

deletes all files starting with AB.

```
PER [QAB*]
```

prints the directory of all filenames which start

with AB.

PER [M/1]

Change default drive (used for GETDISK and PUTDISK) to drive 1.

PER [KDATUMS 6]

erase the disk, and format it with the title "DATUMS" and interleaving 6. Note: Use CTRL+Q to stop/start long directory listings. CTRL+W doesn't work with the PERSCI COMMAND.

PIXEL
Buzzword

is the smallest thing you can change on the screen. The POINT COMMAND will fill one pixel with a COLOR. The screen is divided into 26000 pixels (160*100). There are 160 PIXELs horizontally and 100 vertically. The center of the screen is (0,0) and the PIXELs are numbered -79 to 80 horizontally (X direction) and -49 to 50 vertically (Y direction). The POINT FUNCTION will give you the COLOR VALUE of a pixel.

PLOP
Buzzword

means that whatever COLOR you write with (00-11) will cover whatever is on the screen. So:

Y PLOP X equals Y

For Example:

00 PLOP 10 equals 00

10 PLOP 11 equals 10

PLOP table using 2 BITs.

	PLOP with:			
	0	1	2	3
PLOP				
00	00	01	10	11
01	00	01	10	11
10	00	01	10	11
11	00	01	10	11

The PLOP COLOR MODES are 0-3. The PLOP DISPLAY MODES are 0,5,10,15,20,25,...,130,135,140,145.

POINT(XCOOR,YCOOR)
Function

returns the value (0-3) of the PIXEL ADDRESSed by the two COORDINATES given. 0 means that COLOR (00) is at that ADDRESS on the screen, 1 means that COLOR (01) is there, etc. If the ADDRESS is outside the current WINDOW area then a -1 is returned.
Example:

```
BOX 0,0,20,20,1
PRINT POINT(9,7)
```

prints 1

```
PRINT POINT(25,35)
```

prints 0

POINT XCOORDINATE,YCOORDINATE,COLORMODE
Command

draws a point at XCOORDINATE,YCOORDINATE in the COLOR MODE specified. A POINT is one PIXEL in size. See COLOR MODE.
Examples:

```
POINT 40,15,1
```

draws a red point at 40,15

```
POINT -40,20,2
```

will draw a green point at -40,20.

```
SPIRAL=[angle=0;radius=0
x=radius*sin(angle)
y=radius*cos(angle)
POINT x,y,1
angle=angle+.3
radius=radius+.5
sk -5]
```

SPIRAL draws a spiral made of points starting at 0,0. Press CTRL+C to stop it.

POINT.SNAP (SNAPNAME,XCOOR,YCOOR)
Esoteric Function

returns the value (0-3) of a PIXEL in SNAPNAME ADDRESSed by XCOOR and YCOOR. XCOOR and YCOOR are relative to the center (0,0) of the SNAP. -1 will be returned if the PIXEL is outside the SNAP.
Example:
create a three color SNAP using the macro PSNAP

```
PSNAP=[CLEAR
BOX -10,0,10,20,3
BOX 0,0,10,20,2
BOX 10,0,10,20,1
SNAP FLAG,0,0,30,20]
```

use TEST to find the color value of the PIXELs in the SNAP FLAG

```
TEST=["INPUT X,Y POSITIONS IN SNAP"
INPUT x,y
PRINT POINT.SNAP(FLAG,x,y)
SKIP -3]
```

POINT.SNAP SNAPNAME,XCOOR,YCOOR,COLORMODE
Esoteric Command
changes the PIXEL at XCOOR,YCOOR in SNAPNAME in the COLORMODE specified. XCOOR and YCOOR are relative to the center of the SNAP which is 0,0. You have to DISPLAY the SNAP to see the changes, of course.
Example:
Use the MACRO PSNAP (see previous example) to make the 3 color SNAP called FLAG. The following MACRO will change any of the PIXELs in FLAG which are red (COLOR 01) to blue (COLOR 11).

```
CHANGE=[y=-(FLAG(1)/2);sy=y
x=FLAG(0)/2
sx=-(x-1)
c=POINT.SNAP(FLAG,sx,sy)
IF C==1,POINT.SNAP FLAG,sx,sy,3
IF (sx=sx+1)<=x,skip -2
DISPLAY FLAG,0,0,0
IF (sy=sy-1)>y,sk -5]
PSNAP
CHANGE
```

The X size of FLAG is stored in FLAG(0). The Y size is stored in FLAG(1). This information is used to determine the setup for two nested loops which will go PIXEL by PIXEL through the SNAP FLAG looking for 01 PIXELs and changing them to 11. After an entire horizontal line of the SNAP has been evaluated FLAG will be DISPLAYed.

PORT(NUMBER)
Esoteric Function
returns the VALUE read at the PORT NUMBER identified.
Example:

```
PR PORT(20)
```

will print the value of the switches 0-7. If switches 0,1,2,3 are down, 15 will be printed.

PORT NUMBER1,NUMBER2
Esoteric Command
writes NUMBER2 to the PORT identified by NUMBER1.
Example:

```
TONE=[PORT 22,15
A=A+1
PORT 17,A
SKIP -2]
```

will cause a whooping tone. To stop the tone, press CTRL+C.

PORTS
Idiosyncrasies
are hardware ADDRESSes for DEVICEs and various input and output gadgets. Some are massaged and put into DEVICE VARIABLES (like the JOYSTICKS & COLOR VARIABLEs). Some are accessed with COMMANDS (like RS232 and MUSIC). Of the PORTS listed below only 10,38,39,18,and 19 are likely to

be useful.
Example:
```
     PRINT PORT(N)
```
will print what the value is at PORT N is.
```
     PORT N,K
```
will set PORT N to the VALUE in K

OUTPUT PORTS: (write only)

PORT #:	FUNCTION:
10	Vertical Blanking Line
12	Magic Register
16	Master Oscillator
17	Tone A Frequency
18	Tone B Frequency
19	Tone C Frequency
20	Vibrato
21	Tone C VOLUME and Noise Modulation Control
22	Tone B Volume and Tone A Volume
23	Noise Volume
25	Expand Register
40	Controls Tape Motor Switch 1=on 0=off Bit 0=Motor 1, Bit 1=Motor 2

INPUT PORTS: (read only)

Bit values for each Joystick
Bit 0 UP on (Y)
Bit 1 Down on (Y)
Bit 2 Left on (-X)
Bit 3 Right on (X)
Bit 4 Trigger
Bits 5-7 not used

16	Joystick 1 ($X1,$Y1,$T1)
17	Joystick 2 ($X2,$Y2,$T2)
18	Joystick 3 ($X3,$Y3,$T3)
19	Joystick 4 ($X4,$Y4,$T4)
28	Knob Joystick 1 ($K1)
29	Knob Joystick 2 ($K2)
30	Knob Joystick 3 ($K3)
31	Knob Joystick 4 ($K4)
46	Tablet data

INPUT/OUTPUT PORTS: (read and write)

32	Terminal RS232 data
33	Accessory RS232 data
34	Terminal RS232 control
35	Accessory RS232 control
36	NCUDAT(9511 chip) data port
37	NCUCOM(9511 chip) status port
41	Tape Data Bit 0=Data on input Output toggles port

POWER(NUMBER1,NUMBER2)
Function

returns NUMBER1 raised to the POWER of NUMBER2.
Example:

```
PRINT POWER(5,3)
```

prints 124.999 which happens to illustrate slippage in FLOATING POINT calculations as well.

PRECEDENCE
Buzzword

is the pecking order for the evaluation of OPERATORS in EXPRESSIONs. See OPERATOR for the PRECEDENCE order in ZGRASS.

PRINT THING
Command

THING (a NUMBER, ARRAY VALUE, EXPRESSION etc.) is converted to a STRING if possible and printed followed by a NEXTLINE. Several STRINGs can be used. If you separate them by commas, a space is printed between them. If you do not want the space, separate them with &'s. Stuff in quotes can also be used (like PRINT "THE ANSWER IS:",A). PRINTS (and PROMPTS) are suppressed if there are ARGUMENTs passed to the MACRO.
Examples:

```
PRINT 5
```

will print 5

```
A=1;B=333
PRINT A&B
```

will print 1333

```
ME=7
PRINT 5,ME
```

will print 5 7

```
PRINT "A"&"B"&"C"
```

will print ABC

```
PRINT "FOOT("&1&")="&"BIGTOE"
```

will print FOOT(1)=BIGTOE

PRINT.FORCE THING
Command

like PRINT but forces printing whether or not an ARGUMENT list is passed to the MACRO.
Example:

```
FLUBB=[
PRINT.F "I WAS FORCED TO PRINT THIS"
INPUT n]
FLUBB 4
```

will print "I WAS FORCED TO PRINT THIS" even though you directly passed the value 4.

PRIORITY WRITE
Idiosyncrasy

means a COLOR 00-11, will write over another COLOR if it is greater than or equal to that COLOR.
So: X PRIORITY Y equals MAX(X,Y)
For example:

```
10 PRIORITY 11 equals 11
01 PRIORITY 00 equals 01
```

ZGRASS has two PRIORITY WRITE COLOR MODES 16 and 17. The PRIORITY DISPLAY MODES are 4,14,24,...,124,134,144. See REVERSE PRIORITY.

PRIORITY 16 17

WRITE	00	01	10	11
00	00	01	10	11
01	01	01	10	11
10	10	10	10	11
11	11	11	11	11

PROMPT THING
Command

just like PRINT but does not print the NEXTLINE at the end.

PROMPT.FORCE THING
Command

like PROMPT but forces printing whether or not an ARGUMENT list is passed to the MACRO.

PUNCTUATION
Buzzword

is any typed character which is not a letter, or number, or '$'. Many PUNCTUATION symbols are OPERATORS.

PUTDISK FILENAME
Command

puts FILENAME on disk. FILENAME may be a MACRO, ARRAY or 4K screen dump. If you PUTDISK a FILENAME which is not listed in USEMAP a 4K dump of the screen will be put on the disk. See PERSCI for other disk-related commands.

PUTTAPE NUMBER,FILENAME,STRING
Command

puts FILENAME (MACRO, STRING, ARRAY, SWAP MODULE, Screen dump) on the tape the number of times indicated by NUMBER under the NAME of "FILENAME". If you PUTTAPE a FILENAME which is not listed in USEMAP a 4K dump of the screen will be put out on tape under that FILENAME. The last ARGUMENT is a message to be put with the tape directory header which will print back when scanning the tape with GETTAPE (with CTRL+N) (See GETTAPE). The reason for printing a file several times is to safeguard against errors. An error detection code is stored with each entry on tape and if an error is detected by GETTAPE, it will try the next copy automatically for you.
Example:

PUTTAPE 3,PARMESAN,[THIS IS A SNAP OF CHEESE]

puts out the ARRAY PARMESAN 3 times on tape with the message indicated.

RADIANS
Esoteric Buzzword

PI RADIANS is defined as equal to 180 degrees. One degree is equal to 3.14159/180 RADIANS. One RADIAN equals 180/3.14159 degrees. SINE, COSINE, and TANGENT take values in RADIANS. ARCCOS, ARCTAN, and ARCSINE return values in RADIANS.

RANDOM
Buzzword

is a way of choosing a NUMBER in a range so that the NUMBER is not predictable. The RANDOM OPERATOR in ZGRASS is '%'. 10%100 means pick a NUMBER between 10 and 100 (but not including 100). Each time the % OPERATOR is used, the answer should be different, because it is RANDOM, although sometimes it's the same.

RECURSION
Buzzword

see RECURSION.

RELATIONAL OPERATOR
Buzzword

returns the value of 1 if the condition is true, 0 if false. RELATIONAL OPERATORs are used in IF statements mostly but can be used in other contexts as well since they are OPERATORS just like the arithmetic ones.
The RELATIONAL OPERATORS in ZGRASS are:

OPERATOR:	MEANING:
==	equals
<	less than
>	greater than
<= or =<	less than or equals
>= or =>	greater than or equals
# or <>	not equals

See IF COMMAND for examples.

REPLACE(BIGSTRING,OLDSTRING,NEWSTRING,NUMBER)
REPLACE(BIGSTRING,OLDSTRING,NEWSTRING,NUMBER,LOWER,UPPER)
Esoteric Swap Function

Search for the occurence of OLDSTRING in BIGSTRING from the beginning of BIGSTRING and replace OLDSTRING with NEWSTRING. NUMBER specifies how many times to attempt replacement. The string with the replacement is returned. BIGSTRING is not modified. The matching of OLDSTRING is accomplished in the same manner as in the MATCH routine. You can use expression symbols, as described in the MATCH FUNCTION. If LOWER and UPPER are present they indicate the start location to search and the end location. If UPPER is less than LOWER the search is done backwards (that is, from UPPER down one by one to LOWER).

Examples:

```
PR REPLACE("ABA","A","-*-",1)
```

prints out "-*-BA"

```
PR REPLACE("ABA","A","-*-",1,5,0)
```

prints out "AB-*-"

```
PR REPLACE("SUNSHINE","SUN","MOON",3)
```

prints out "MOONSHINE"

```
PR REPLACE("UNIVERSITY OF ILLINOIS AT CHICAGO
CIRCLE","*","UICC",1)
```

prints out "UICC"

```
PR REPLACE("THIS IS A VERY EASY
TEST","[AEIOU]","-",20,10,0)
```

prints out the string "TH-S -S - VERY EASY TEST"

```
NOISE=[BEEP THE JEEP]
PR REPLACE(NOISE,"EEP","UNK",2)
```

prints out "BUNK THE JUNK". NOISE is unchanged.

RESOLUTION
Buzzword

is the measure of the number of PIXELS on the TV screen. The RESOLUTION of ZGRASS is 160 by 100.

RESTART
Command

clears memory and restarts ZGRASS. This is a software way to push the RESTART button, and you must answer "Y" to RESTART when it asks. The "N" option is there since system failure often results in automatic restarts, and typing "N" in prevents you from losing everthing in MEMORY.

RETURN
Command

returns control to the calling MACRO. Same as running off the end of a MACRO.

RETURN VALUE
Command

returns the VALUE indicated and control to the calling MACRO. Useful for creating user defined FUNCTION calls which return values.
Example:

```
MAX=[INPUT a,b,c;.NOTE THE local VARIABLES
IF a<b,IF b<c,RETURN c
IF a>b,IF a>c,RETURN a
RETURN b]
```

This will return the maximum of the three parameters passed and could be used in:

```
BIGGEST=MAX(OF,THESE,THREE)
HONEY=MAX(CRUNCH1,CRUNCH2,KISS)
NUWAVE=MAX(?a,?b,?c)
```

The last is an example of passing LOCAL VARIABLEs.

REVERSE-PRIORITY
Buzzword

means a COLOR 00-11, will write over another COLOR if it is less than or equal to that COLOR.
So:

X REVERSE PRIORITY Y equals MIN(X,Y)

For example:

10 REVERSE PRIORITY 11 equals 10
10 REVERSE PRIORITY 01 equals 01

ZGRASS has two REVERSE PRIORITY COLOR MODES 19 (01-red), and 18 (10-green). See PRIORITY WRITE.

REVERSE PRIORITY	19 00	19 01	19 10	19 11
00	00	00	00	00
01	00	01	01	01
10	00	01	10	10
11	00	01	10	11

RS232(NUMBER)
Esoteric Function

returns the INTEGER value of the RS232 PORT indicated by NUMBER. If NUMBER==0, the terminal is read, if NUMBER==1, the accessory RS232 PORT is read. 0 is returned if no character is at the PORT.
Examples:

```
GETAKEY=[PRINT "PRESS A KEY"
A=RS232(0)
IF A==0,SK -1
IF A>47&&A<58,PRINT "YOU PRESSED THE ",A-48,"
KEY"]
```

(Note: this will not work well in .B or .F MACROS because key presses are automatically sent to "calculator mode.")

```
ANYBODYTHERE=[A=RS232(1)
IF A#0,PRINT "WAKEUP"]
```

This will print "WAKEUP" if a device or other computer is trying to talk to ZGRASS over the accessory PORT. Note: since 0 indicates no character, you cannot receive an ASCII null character. Also note that the high bit of each character is set to 0 automatically.

RS232 NUMBER1,NUMBER2
Esoteric Command

If NUMBER==0, then write to the terminal. If NUMBER1==1, then write to the accessory RS232 PORT. NUMBER2, a VALUE from 0-255, is written to the PORT chosen.

Example:

```
RS232 0,7
```

will make the terminal beep. A table of ASCII values in decimal will help you with this COMMAND.

Note: you can transmit an ASCII null character by:

```
RS232 1,0
```

SCALE XSCALE,YSCALE,SNAPNAME,XCENTER,YCENTER,DISPLAYMODE
Swap Command

takes SNAPNAME and scales it on its X and Y axes using XSCALE and YSCALE, and then writes it to the screen at XCENTER,YCENTER with the specified DISPLAYMODE. The range for XSCALE and YSCALE is -128.00 to 127.00. A negative scale factor will give a mirror image. SCALing by 1 on both axes will give the original SNAP. SCALing by 0 will result in ERROR #24.

Example:

Bring in from disk or tape the Swap Commands SCALE and TXT, and connect up JOYSTICK #1.

```
LARGE=[CLEAR
TXT 0,0,1,1,1,2,"FROG"
SNAP RRR,12,3,26,7
X=8;Y=8
IF X#0,SCALE X,Y,RRR,0,0,0
IF $T1==0,SK 0
IF (X=X-1)>-8,Y=X,SKIP -2]
LARGE
```

A SNAP called RRR is made of the word FROG by writing it on the screen using the TXT Command. Press the trigger to continue. The next image you see is a SCALEd version of the SNAP RRR 8 times the size of the original. By pressing the triggev you$get RRR SCALEd by 7. This will continue on from 6,5,4,...,-6,-7. By SCALing with a negative number you can reverse your image.

SCROLL XCEN,YCEN,XSIZE,YSIZE,XMOV,YMOV,DISPLAYMODE,FCOLOR
Swap Command

moves an area of the screen centered at XCEN,YCEN of XSIZE, YSIZE dimensions with DISPLAYMODE in the direction defined by XMOV,YMOV using FCOLOR to fill in the old area. FCOLOR can be any one of the 4 COLORs 0-3.

SEMANTICS
Buzzword

The meaning of a COMMAND as opposed to its SYNTAX (the rules for how it is to be spelled, punctuated, etc.)

SHRINK XSCALE,YSCALE,NAME,XCENTER,YCENTER,XSIZE,YSIZE
Swap Command

is like SNAP but shrinks or expands the part of the screen it is SNAPping. Only positive VALUEs for XSCALE and YSCALE will work.
Example:

```
BOX 0,0,160,100,1
BOX 25,0,20,100,2
BOX 0,25,160,30,6
SHRINK .25,.3,SCREEN1,0,0,160,100
CLEAR
DISP SCREEN1,0,0,0
SHRINK .25,.3,SCREEN2,0,0,160,100
DISPLAY SCREEN2,0,0,0
CLEAR
DISP SCREEN1,-25,0,0
DISP SCREEN2,25,0,0
```

SINE(NUMBER)
Function

returns the sine of NUMBER.

SKIP NUMBER
Command

skips the given NUMBER of lines (including the one you are on). It transfers control by counting the NUMBER of NEXTLINE's indicated. SKIP 0 hangs in place, SKIP 2 skips the next 2 lines, SKIP -3 goes back 3 lines. SKIP 999 is the same as RETURN and SKIP -999 will get you back to the beginning of the MACRO. SKIP does not allow LABELs. Use GOTO with LABELs.
Examples:

```
SKIP 0;.GOES TO THE BEGINNING OF THIS LINE
SKIP 2;.SKIPS THE NEXT TWO LINES
SKIP -3;.GOES BACK 3 LINES
SKIP 1;.GOES TO THE NEXT LINE

TOGO=[m=10
PRINT m,"TO GO"
IF (m=m-1)>0,SKIP -1
PRINT "NO MORE"]
```

SNAP NAME,XCENTER,YCENTER,XSIZE,YSIZE
Command

takes the PIXELs in the area indicated and saves them in an ARRAY called NAME. The DISPLAY COMMAND is used to redraw the ARRAY somewhere else. The SCALE COMMAND is used to scale and redraw the ARRAY somewhere else. NAME(0) gets the XSIZE and NAME(1) gets the YSIZE for your use. Example:

```
FLASH=[s=14
BOX 0,0,s,s,5%8;IF (s=s-1)>1,SK 0
SNAP ART,0,0,16,16
DISP ART,x=x+$X1,y=y+$Y1,0;SKIP 0]
FLASH
```

FLASH will draw some BOXes, make a 16X16 SNAP called ART, and finally allow the user to move the SNAP around on the screen using JOYSTICK #1.

SNAPPED PIX
Buzzword

is a special ARRAY which contains PIXELs from an area on the screen specified by a SNAP COMMAND. See SNAP and DISPLAY.

SQRT(NUMBER)
Function

returns the square root of NUMBER.

STATUS 0,XCENTER,YCENTER
Swap Command

returns the X,Y COORDINATES of the current center of the screen in XCENTER,YCENTER. See CENTER.

STATUS 1,LEFTX,BOTTOMY,RIGHTX,TOPY
Swap Command

returns two X COORDINATES and two Y COORDINATES which describe the boundaries of the current WINDOW. See WINDOW.

STOP NAME
Command

is used to selectively halt the EXECUTion of a MACRO or COMPILEd MACRO running in .B or .F mode. A MACRO/COMPILEd MACRO can stop itself or any other MACRO.

STRING
Buzzword

is a collection of characters (numbers, letters, punctuation) delimited (enclosed) by single or double quotes or balanced square '[]' or curly '{}' brackets. If you have to use a string delimiter in a STRING, make sure it is delimited by a different string delimiter or things will get very confused (most likely it will consider the rest of your MACRO as part of the STRING). Examples:

```
"THIS IS A STRING"
"PRINT A*B*C
SKIP -1;.THIS STRING COULD BE A MACRO TOO"
[1234]
PRINT ['];.A QUOTE IN A STRING
```

STRING(NAME,NUMBER)
Esoteric Function

returns the INTEGER which represents the character in the position indicated by NUMBER. Can be used to access STRINGs as BYTE ARRAYs.

Example:

```
TYPE=[
PRINT "INPUT A STRING OF CHARACTERS"
INPUT.STRING CHAR
A=0
B=STRING(CHAR,A)
IF B==0,SK -4
PRINT B,"IS ASCII FOR",ASCII(B);A=A+1;SK -2
SK -6]
```

This prints out the decimal ASCII values of the string of characters which you input and are stored in CHAR. If you inputted "ABC" you should get 65,66,67. The listing of characters stops when it encounters the null (INTEGER value 0) at the end of CHAR (and every STRING).

STRING NAME,NUMBER1,NUMBER2
Esoteric Command

puts NUMBER2 into the STRING "NAME" offset by the number of BYTES in NUMBER1.

Example:

```
LETTERS="ABCDE"
STRING LETTERS,3,50
PRINT LETTERS
```

will print ABC2E

Note: allowing NUMBER1 to exceed the length of the STRING can clobber innocent MEMORY and lead to system failures. You can use a STRING as a BYTE ARRAY only if you have first made it large enough by CONCATENATION or ASSIGNMENT. This command and the ASCII command are potentially useful for communications over the accessory RS232 PORT.

STRING VARIABLE
Buzzword

is a NAME that has a STRING as its VALUE.

SUBSTR(MYSTRING,BEGIN,END)
Esoteric Swap Function

returns a STRING value that is the subset of MYSTRING specified by the BEGIN and END displacement values. If the END value extends beyond the end of MYSTRING, the substring simply contains all the characters of MYSTRING following BEGIN. A null string is returned if the value of BEGIN extends beyond the end of MYSTRING.

Examples:

```
PR SUBSTR("ABCDEF",0,2) prints out the string
"ABC"
PR SUBSTR("ABCDEF",4,20) prints out the
string "EF"
```

SWAP COMMAND or FUNCTION
Idiosyncrasy
is a COMMAND or FUNCTION written in assembly language which must first be gotten into memory from disk or tape.

SWAP MODULE
Idiosyncrasy
Certain segments of system code (specifically extended graphics and music) are on tape on disk. They must be EXECUTEd to link them into the system code so the features are available. See MUSIC and COLORS.

SWITCH
Buzzword
is an option for COMMANDS, FUNCTIONS, and MACROS. The only switches defined for MACROs are .B and .F which cause the MACRO to be EXECUTED in the background and foreground respectively. Many COMMANDS and FUNCTIONS (INPUT, ARRAY, etc.) have SWITCHes which are given as separate entries in this glossary. SWITCHes are always preceded by the NAME they are modifying and a '.'.
Examples:
```
INPUT.STR SAM
ARRAY.INT FOO,123
DEATHWEAPON.B
```

SYNTAX
Buzzword
is the form of a language, its spelling, punctuation, words, etc. (Contrast with SEMANTICS.)

TABLET X,Y
Swap Function
returns the X,Y values of the TABLET pen position in X and Y, and the value of the pen push (0=not pushed but on surface, 1=pushed, -1=off surface), to the caller.
Example:
```
PXY=[A=TABLET(X,Y)
X=X/12;Y=Y/12
IF A==1,BOX X,Y,2,2,3
IF A==0,BOX X,Y,4,4,5;BOX X,Y,4,4,5
SKIP -4]
```
This will put a blue BOX(if the pen is pushed), or a flashing red BOX at the pen's current location. The X and Y values' range is:
$-1100 < X,Y < 1100$
Of course, any VARIABLE NAME can be used instead of X and Y.

TANGENT(NUMBER)
Function
returns the tangent of NUMBER.

TERMINAL
Esoteric Command
TERMINAL bypasses the keyboard and the CRT directly in connection with the accessory RS232 PORT so you can connect up to another computer system as a terminal. BREAK gets you back to ZGRASS.

TERMINAL ARG0,ARG1,...ARG9
Esoteric Command
allows user to specify one of three terminals with ARG0. Set ARG0 to 0 for Hazeltines, 1 for ADM3As, 2 for ACTIVs. Then, up to 9 decimal ARGUMENTS may be entered. ARG1 allows you to define an additional key for rubout outside EDIT (ESC or underscore work well). ARG2-9 specify the EDIT keys:

	Maps into:	
ARG2	CURSOR Right	08 (^H, Backspace)
ARG3	CURSOR down	09 (^I, Tab)
ARG4	CURSOR Up	010 (^J, Linefeed)
ARG5	CURSOR Left	011 (^K)
ARG6	INSERT char	94 (^)
ARG7	Delete Char	18 (^R, HOME)
ARG8	Delete Line	01 (^A, CLEAR)
ARG9	Extra for RUB	127 (RUBOUT)

Examples:

```
SETUP=[TERMINAL 2,95,8,11,26,24,94,9,27,95]
SETUP
```

Sets up for an ACTIV using underscore as an alterative for DEL (rubout) both in and out of EDIT. It also specifies the arrow keys for cursor left, right, up, and down in EDIT. Delete character, in this example is TAB and delete line is ESC. You need an ASCII table for your terminal to use this command successfully.

```
ADM3A=[TERMINAL 1,95,12,10,11,8,94,18,1,95]
ADM3A
```

Sets up for an ADM3A.

TEXT SPACARRAY,XLEFT,YLOWER,DMODE,HORSP,VERSP,TSTRING,
FONTARRAY1,FONTARRAYN
Esoteric Swap Command

is used to generate strings of text or arbitrary figures on the TV screen. The size of the text, the styles, the colors, and spacing are all user definable through the FONT COMMAND and the TEXT COMMAND itself.

ARGUMENT: Description:

SPACARRAY if default (no special spacing) wanted, use XXXX (or some other NAME you're not using), otherwise specify a text spacing ARRAY as described in FONT.

XLEFT is the X COORDINATE where TSTRING is to begin.

YLOWER is the bottom row of PIXELs on which TSTRING is to be displayed.

DMODE is the DISPLAY MODE. Any ZGRASS DISPLAY MODE can be used.

HORSP is any positive or negative INTEGER or zero. It represents a constant spacing factor in PIXELs to be inserted between characters.

VERSP is an INTEGER which signifies the number of pixels to move up (-) or down (+) on seeing a NEXTLINE in TSTRING.

TSTRING is the STRING to be displayed. Every character in the STRING should have been previously defined in a FONT ARRAY named in the next operand. If a character isn't found in one of the named ARRAYs, the character is ignored, and no warning is given.

FONTARRAY1,FONTARRAYN are the NAMEs of the FONT ARRAYs to be used. The ARRAYs are searched in the order given. The number of ARRAYs that can be entered is only limited by the number of characters you can type on a line.

TIMEOUT NUMBER
Esoteric Command

wait for NUMBER/60 seconds and then return.
Example:

```
FOO=[TIMEOUT 300
PRINT "5 SECONDS UP"]
FOO.F
```

Every 5 seconds "5 SECONDS UP" will be printed. Works only with .F.

TRUTH TABLES
Buzzword

See AND, OR, PLOP, PRIORITY, REVERSE-PRIORITY and XOR.

TXT X,Y,XSIZE,YSIZE,FCOLOR,BCOLOR,DISPLAYMODE,CHARSTRING
Swap Command

prints CHARSTRING on the TV screen starting at X,Y with the character size specified by XSIZE,YSIZE, in FCOLOR with BCOLOR as the background COLOR using the specified DISPLAYMODE. The smallest values for XSIZE,YSIZE are 1,1 which means that the characters will be 5 PIXELs wide and 7 PIXELs high. The largest character can take up 4K or the largest available chunk of memory.
Examples:

```
TXT 0,0,1,1,1,0,"SMALLTEXT"
```

this will print "SMALLTEXT" starting at 0,0 with 5X7 characters in red (01) with a white background (00).

```
TXT -50,-15,2,2,3,1,"SMALLTEXT * 2"
```

will print "SMALLTEXT * 2" starting at -50,-15 with 10X14 characters in blue (11) with a red background.

USEMAP
Command

gives a list of NAMEs currently in use and the number of BYTEs they take up.

VALUE
Buzzword

is a NUMBER or STRING typically. PRINT will always tell you the value of a CONSTANT or a VARIABLE.

VARIABLE
Buzzword

is a NAME you can use to hold a VALUE. Any NAME in ZGRASS that can be put on the left side of a '=' is a VARIABLE and its VALUE can be varied by that ASSIGNMENT (which is why it's called a VARIABLE instead of a CONSTANT, of course). USEMAP will give you information about your VARIABLES. Note: that VARIABLES A-Z are built into the system and are not listed in USEMAP.

WAIT NUMBER
Command

waits the specified NUMBER of seconds before continuing by doing a SKIP 0 until the time is up.
Example:

```
NEST=[A=0
A=A+10
BOX 0,0,A,A,7
WAIT 2
IF A<100,SK -3]
```

This will draw a BOX waiting approximately 2 seconds before starting another. To wait a fraction of a second, use a System Timer which counts in 1/60 seconds.

```
TENPERSECOND=[$ZO=6
IF $ZO#0,SK 0
PR "XX"
SK -3]
```

this will print "XX" ten times per second

WHATSIS (NAME)
Esoteric Swap Function

returns an INTEGER value for the type represented by the NAME.

```
  Values:     Meaning:
    2   ;Null NAME
    8   ;STRING NAME
    14  ;NUMBER NAME
    16  ;ARRAY NAME
    18  ;COMPILEd MACRO NAME
    20  ;SWAP MODULE
```

Example:

```
MACROONLY=[GETTAPE SUE
A=WHATSIS(SUE)
IF A#8, SK -2]
```

This will set A to SUE's type. If you PUTTAPEd a SUE that was a SNAP and a SUE that was a MACRO, waiting for A to equal 8 would allow you to skip the SNAP called SUE.

WINDOW XLEFT,YBOTTOM,XRIGHT,YTOP
Command

creates a window in the ZGRASS screen with XLEFT as the left side, YBOTTOM as the bottom side, etc. CLIPPING is done for all drawing COMMANDS. Windows are CLIPPED to the screen and use the same COORDINATE system unless changed by CENTER. WINDOW.FULL resets the WINDOW to full screen.
Example:

```
CLEAR
WINDOW -20,-30,20,30
VIEW=[BOX -79%80,-49%50,8,8,5%8
SKIP -1]
```

WINDOW.BOX XCENTER,YCENTER,XSIZE,YSIZE
Command

is the same as WINDOW except you specify it like BOX using XCENTER,YCENTER to mark the center and XSIZE,YSIZE to specify the dimensions of the WINDOW.

WINDOW.FULL
Command

resets the WINDOW to full screen.

WRAP XCEN,YCEN,XSIZE,YSIZE,XMOVE,YMOVE
Swap Command

moves an area of the screen centered at XCEN,YCEN of XSIZE,YSIZE dimensions in the direction defined by XMOVE,YMOVE around onto the originally defined area by wrapping around.

WRAP AROUND
Buzzword

is the phenomena that causes OVERFLOWed VARIABLES to print as weird numbers. If a DEVICE VARIABLE OVERFLOWs at 255, 256 WRAPS AROUND to 0, 256 to 1, etc. This is the same as modulus arithmetic with base 256.

XOR
Buzzword

is a LOGICAL operation (also called 'exclusive or') used to draw PIXELs on the screen. What gets drawn is a value from 0-3 and is computed by the XOR function of what there was on the screen with what you give it to write there. The reason for this complexity is that a couple of neat tricks are made possible by XOR. First, if you draw anything on the screen with XOR (COLOR MODES 4-7) or DISPLAY a SNAPped picture element with DISPLAY MODE 1, you can erase it by simply drawing or DISPLAYing it again the same way. In other words, two XOR's is the same as nothing. Second, by setting $L3=$L2 (and $R3=$R2 if you mess with $HB), you can make anything written with COLOR 1 pass 'behind' anything written with COLOR 2 (you have to try it to believe it). XOR table using 2 BITs.

XOR	4	5	6	7
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

The XOR COLOR MODES are 4-7. The XOR DISPLAY MODES are 1,6,11,16,21,26,...,131,136,141,146.

XR NUMBER1,NUMBER2
Esoteric Swap Command

ZGRASS uses three stacks to manage subroutining. Normally you can pass about 40 ARGUMENTS and go about 6 levels deep before running out of stack space. The XR COMMAND allows you to reorganize some of your RAM for stack space. NUMBER1 indicates the total number of ARGUMENTS you wish to pass. 800 is the maximum for NUMBER1. NUMBER2 is the level to which you want to nest RECURSIVE subroutines. 120 is the maximum for NUMBER2. The largest values for NUMBER1 and NUMBER2 use up quite a bit of MEMORY. If your RECURSIVE routines exceed the limits you set using XR, the system will most likely print an ERROR message relating to the effects of wanton memory writes. There isn't any ERROR checking on stack overflows in this mode. If you need more stack space, you can use the XR command again but the old space is not reclaimed so it is best to RESTART first. Each MACRO/CPL invocation temporarily uses about 110 BYTEs so if you go 100 deep, 11000 additional BYTEs will be chewed up at the deepest level. So a MACRO like TOM below could easily use half your MEMORY.

Example:

```
XR 400,100
K=0
TOM=[INPUT A,B,C,D;K=K+1;CORE;WAIT 1
IF K<100,TOM 1,2,3,4]
```

Notice how CORE is decreasing as your MACRO is EXECUTing.

ZAP1(INTEGER)
Swap Function

takes the INTEGER as an ADDRESS and returns a 8-BIT value.

Example:

```
DUMP=[A=0
PRINT ZAP1(A)
A=A+1
IF A<16384,SK -2]
```

This will print a decimal dump of the ZGRASS code for anyone who is into machine code disassembling.

ZAP1 INTEGER1,INTEGER2
Swap Command

puts INTEGER2 (8-BIT VALUE) into the space addressed by INTEGER1.

Example:

```
STRIPES=[A=16384
ZAP1 A,A\80
A=A+1
IF A<32768,SK -2]
```

This will print multicolored stripes by directly addressing the screen. Be careful, this command can wipe out (ZAP) the system.

ZAP2(INTEGER)
Swap Function
takes the INTEGER as an ADDRESS and returns a 16 BIT value.
Example:

```
PRINT ZAP2(0)
```

prints out -1085

ZAP2 INTEGER1,INTEGER2
Swap Command
puts INTEGER2 into the space ADDRESSed by INTEGER1.
Example:

```
ZCOR=[A=0
ZAP2 A+16384,ZAP1(A)
A=A+2
IF A<32768,SK -2]
```

This will dump the ZGRASS machine code in ROM into the screen MEMORY for you to see.